공학박사 학위논문

An Efficient Analysis of Resin Transfer Molding Process Using eXtended Finite Element Method

확장 유한요소법을 적용한 Resin Transfer Molding 기법의 효율적 수치해석

2013 년 2월

서울대학교 대학원

협동과정 계산과학전공

정 연 희

An Efficient Analysis of Resin Transfer Molding Process Using eXtended Finite Element Method

확장 유한요소법을 적용한 Resin Transfer Molding 기법의 효율적 수치해석

지도 교수 김 지 환

이 논문을 공학박사 학위논문으로 제출함
2012 년 12월

서울대학교 대학원
협동과정 계산과학전공
정 연 희

정연희의 공학박사 학위논문을 인준함
2012 년 12월

위 원 장 _____ (인)

부위원장 _____ (인)

위 원 _____ (인)

위 원 _____ (인)

위 원 _____ (인)

ABSTRACT

Yeonhee Jung

Interdisciplinary Program in Computational Science and Technology

The Graduate School
Seoul National University

Numerical simulation for Resin Transfer Molding (RTM) manufacturing process is attempted by using the eXtended Finite Element Method (XFEM) combined with the level set method. XFEM allows to obtaining a good numerical precision of the resin pressure near the resin flow front, where the gradient of the pressure is discontinuous. The enriched shape functions of XFEM are derived by using the level set values so as to correctly describe these shape functions with the resin flow front. In addition, the level set method is used to transport the resin flow front at each time step during the mold filling. The level set values are calculated by an implicit characteristic Galerkin FEM. The multi-frontal solver of IPSAP is adopted to solve the system. This work is validated by comparing the analysis results with analytic solutions.

Moreover, the localization of XFEM and level set method is proposed to increase the computing efficiency. The computation

domain is reduced to the small region near the resin flow fronts. Therefore, the total computing quantity is minimized by the localization. The efficiency test is made with simple channel and radial flow models. The localization gives notable reduction of the computing time

Several application examples are analyzed to demonstrate ability of this study. The merging and separating of the flow fronts describes through the first analysis. In addition, the race tracking effect and macro void formation during RTM process are simulated. The wind turbine blade are also used to analyze its mold filling during RTM process. Finally, the proposed study is realized with Graphic User Interface (GUI) tool for pre/post-processing of the simulation.

Keywords : RTM Process, XFEM, Level Set Method, Numerical Analysis, Localization,

Student Number : 2005-23272

Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1. Overview

The Resin Transfer Molding (RTM) process is a manufacturing method of composite structures.

(a) Automotive (b) Shipbuilding

(c) Aerospace

Figure 1.1. Applications of the RTM process in the industry of transport

It is widely used in the industry of transport (e.g., automotive, shipbuilding and aerospace) as shown in the figure 1.1. The RTM process offers relatively low cost for manufacturing structures of complex shape and medium/large size. The surface definition and appearance of the structures are good because the RTM is a closed mold process.

(a) Placing the preform

(b) Closing mold

(c) Injection of resin

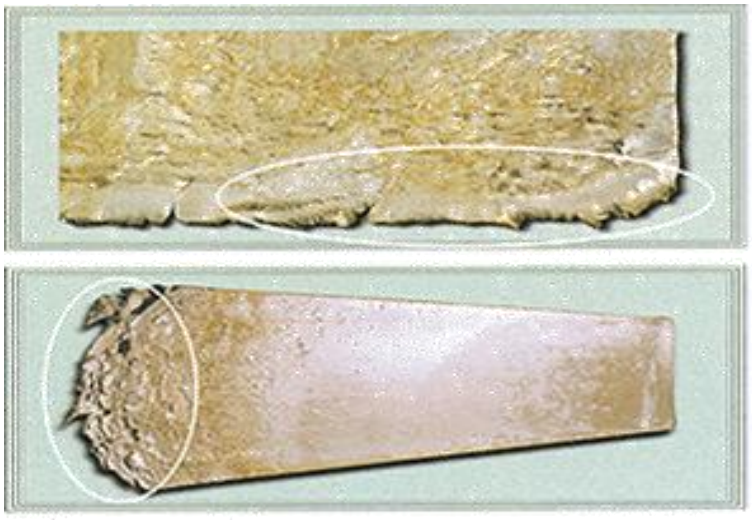
(d) Curing

(e) Demolding

Figure 1.2. Manufacturing procedures of the RTM process

The manufacturing procedures of the RTM process are described in the figure 1.2. The RTM process uses lower and upper molds. The cavity between the molds takes the shape of structure to be produced. The fiber preform is inserted in the mold cavity and compressed by closing the molds. The preform is filled with the resin which is injected through gates from a pressured container. After completely filling the preform, the resin is allowed to cure. The mold is opened when the structure is hardened after the curing.

(a) Formation of air voids



(b) Incomplete filling

Figure 1.3. Defects of structures manufactured by the RTM process

For the manufacturing of composite structures by RTM process, it is important to predict weld lines and to avoid the formation of air voids, which could cause critical defects in the structure. The anticipating defects by incorrect filling of the mold are shown in the

figure 1.3. Manufacturers should rebuild the mold to produce correct structures whenever they find the defects. However, the mold rebuilding is expensive and time-consuming. Furthermore, to set up the mold filling stage during the RTM process, various variables should be considered such as injection pressure, injection gate/vent numbers, and their locations. The design of these variables is sometimes expensive and time-consuming since the motion of the resin flow is impossible to check in a closed mold while filling the mold. A numerical simulation can be applied as a cost and time effective method to complete the RTM process.

During RTM process, a resin flow in a mold is affected by resin viscosity and permeability of fiber preform. The resin viscosity is changed during injection process due to the change of temperature. However, because the filling time is shorter than curing time or the change of temperature is small, the viscosity is assumed to be isothermal and constant during the filling process. Permeability is related with fabric of fiber preform. When the preform is closed fabric, the resin flow slowly moves. On the contrary to this, when the preform is opened fabric, the resin passes fast. On macroscopic scale, the preform is assumed as non-deformable condition, therefore, it is a constant anisotropic value during the mold filling.

The resin flow in the mold represents a moving interface problem since the resin flow front moves through the preform placed in the mold during the filling process. The numerical methods for the moving interface problem could be divided into two groups: moving grid and fixed grid. The fixed grid approach has been more widely used, because the moving grid approach needs relatively expensive and difficult computation.

In the industry, the structures manufactured by RTM process are complex shape with hole and curvature as well as simple plate types. However, the simulation results are inaccurate when applying traditional numerical approaches for analysis of the complex shape structures. Even though it is possible to acquire appropriate results by using the very fine mesh models, the computation time becomes longer. Therefore, many researchers have investigated to find approaches to increase the accuracy and efficiency of the RTM process simulation.

1.2. Literature Review

RTM process has become an popular method of manufacturing middle/large structures of fiber reinforced composite materials because of its capabilities including non-expensive process equipment, excellent control on mechanical properties, closed mold process, low filling pressure, and so on, as described in Ref. [1-5]. However, the design of mold filling equipment is quite difficult because the reinforcement consisting of several layers of fiber mats inside the closed mold should be perfectly impregnated by the resin which enters through several injection ports [6-10]. Therefore, many researchers have investigated for numerical simulation of mold filling stage in RTM process [11-16]. Designers have used the simulation to predict pressure distribution, resin flow pattern, void formation, and other phenomena during mold

filling[11-13, 15-16]. Nevertheless, the simulation has been restrictedly utilized because the simulation could not give an accurate prediction of the mold filling stage for manufacturing complex shape structures. Actually, the early researches had analyzed a simple shape structures due to the difficulty to analyze the moving resin fronts [17-20]. Still now, many researchers have investigated to develop more proper methods for the analysis of complex shape structures [14-26].

The Volume of Fluid (VOF) method was popularly used to track the resin flow front for RTM process simulation [17-26]. The VOF method is one of the representative methods to track moving interfaces [27, 28]. However, because the moving interface is defined by the fractional fluid volume in the element or control volume according to the used method, it could be inaccurate for the cases to track complex movements of the resin front [29-33]. The level set method is the representative of methods to capture moving interfaces [30~41]. The level set method is often used for RTM process simulations since it predicts intuitively the flow front's position and is very efficient for multiple flow fronts occurred by complex shapes of the mold or by multiple injection gates [29, 32].

The most frequently used numerical methods during the last two decades are the Control Volume Finite Element Method (CVFEM) and the nonconforming FEM. CVFEM was the most frequently developed numerical method for RTM process simulation until the beginning of the years 2000, because CVFEM could efficiently satisfy the mass conservation of the resin between two adjacent elements, with regard to the standard finite element approach [17-26, 42-46]. However, the construction of control volumes in the whole computational domain could make the computation complex [22-24]. Trochu et al. proposed to

apply nonconforming FEM to the RTM process simulation. Since the nonconforming FEM calculates the pressure at the inter-elements boundaries, the flow rate at the element boundaries can be obtained without any additional treatment, from which the filling factor is derived. However, the filling factor is not the nodal value but the element one [24]. That is why these methods could not properly approximate the pressure field in the elements intersected by the resin flow front because of the discontinuity of the pressure gradient field near the flow front. This point is very important in the case of industrial problems having complex shapes.

The eXtended Finite Element Method(XFEM) is widely used to treat various discontinuity problems such as crack or interface problems [47-62]. The early works about XFEM were focused on the simulation of fracture and went towards the development of XFEM for fracture mechanics [48-53]. Furthermore, in the field of contact mechanics, XFEM techniques for powder compaction problems are presented by Khoei et al. [54]. Moes et al. [55] used XFEM to solve scales with complex geometry in multiscale analysis of components. Chessa et al. [54] presented XFEM for the multi-dimensional Stefan problems. And, Chessa et al. [57, 58] applied XFEM to two-phase immiscible flow problems having a discontinuity of the pressure gradient at the interface between two fluids.

1.3. Scope of Research

One of the important points to simulate the resin flow in the mold is to provide accurate approximation of the moving resin flow front during the mold filling process. Moreover, it is necessary to increase the computational efficiency so as to be able to treat industrial problems within an acceptable computing time.

This study suggests to apply eXtended Finite Element Method (XFEM) combined with level set method to analyze the mold filling stage during RTM process. The main advantage of this approach is the more accurate approximation of the pressure distribution in the mold and resin flow front location at each time step. And, this study attempts the formulation compels the coefficient matrix to be symmetric and positive definite in order to apply the more efficient linear solver. In addition, this study proposes to apply the techniques, such as localization of XFEM and level set method and usage of an effective solver, for increasing the computational efficiency.

The enriched shape functions of XFEM describing the pressure field are defined by the level set values. The pressure field is divided into two regions: normal and enriched. The enriched region is defined only near the resin flow front; therefore, the size of the linear system slightly increases. Since the XFEM formulation done by this study makes the coefficient matrix to be symmetric and positive definite, the multi frontal solver proposed by IPSAP [63, 64], an effective approach for solving the equilibrium systems, is used in order to reduce the computing time.

The level set method is composed of two procedures. The first one is to transport the flow front, and the second is the reinitialization to keep the level set function close to the signed distance function. An implicit characteristic Galerkin Finite Element Method (FEM) is used to stabilize numerical oscillations caused by the convective term in the governing equations. This formulation also makes the coefficient matrix to be symmetric and positive definite.

This study is validated through several examples in comparison with analytical or experimental results found in the literature [22-24, 30, 66]. Additionally, complex shaped plate examples are used to simulate race-tracking effect and the formation of macro voids. The wind turbine blade example is also simulated. The exclusive program for RTM process simulation (Diamond/RTM) is embedded in Diamond program [70], which is an integrated finite element program with GUI.

1.4. Summary

This paper is organized as follows. Chapter 2 explains about numerical methods in order to show the basic knowledge and the difference of the traditional methods, XFEM and level set method. .At first, there is an introduction about CVFEM and nonconforming FEM with VOF method, which are the traditional numerical methods for RTM process simulation. And, the next sections account for XFEM and the level set method which are applied in this study.

Chapter 3 enters into details about the advanced numerical analysis proposed by this work. At first, it is explained about mathematical modeling of RTM process. Next, the numerical formulations by XFEM and level set method are presented. This section is divided by two parts: the formulations for the pressure calculation (XFEM) and the transport of resin flow front (level set method). Furthermore, the whole computational procedures are minutely depicted. Finally, this study is validated by using some examples. Especially, the analysis result by this study is compared with an analytical result.

Chapter 4 goes on to trace the computing efficiency. At first, localization of XFEM and level set method are explained in this chapter. Furthermore, the computational efficiency by the localization is investigated.

Chapter 5 shows the applications of this study. The plate with complex edge is analyzed to show. In addition, the race tracking effect and the macro void formation in the mold are analyzed by the thin and

complex shaped structure. The analysis for manufacture of the wind turbine blade is placed. Finally, the pre and post processors of Diamond/RTM are described.

In Chapter 6, the conclusions of this research are summarized with suggestion for future work.

Chapter 2. Numerical Methods

This chapter introduces numerical methods applied in RTM process simulation. The purpose of this chapter is to show the difference between the traditional numerical methods and the methods used in this work. At first, the conventional approaches for RTM process simulation are introduced. Furthermore, there are explanations of XFEM and the level set method, which are applied for the analysis of RTM process in this study.

2.1. Conventional Approaches

A. VOF Method

VOF method has been frequently used to solve moving boundary problems in a fixed mesh. And, it is the representative of the boundary tracking method [30]. For the numerical simulation of RTM

process, the resin flow front position in the mold is obtained by VOF method [27].

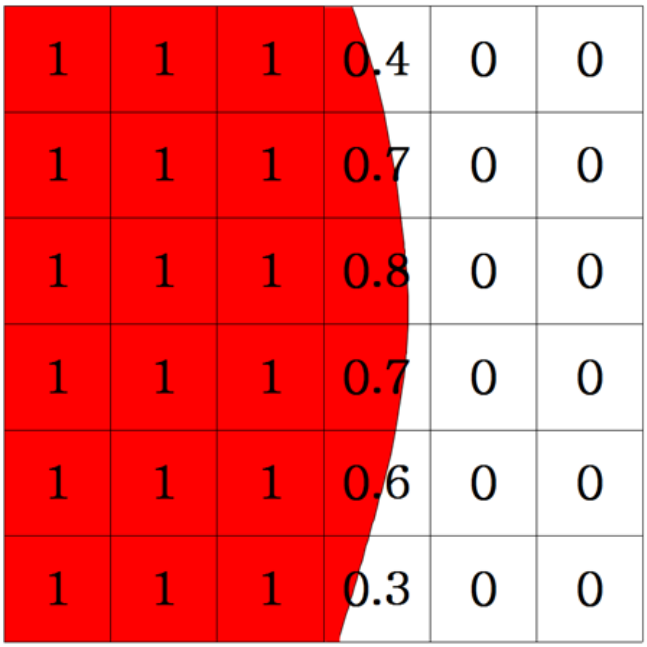


Figure 2.1. Filling factors at filled, front, or empty region
(the red color means the region filled by resin)

In VOF method, the filled or empty region is divided by using a filling factor, which is the volume of fraction in each cell. The figure 2.1 shows filling factors at filled, front, and empty regions. The numbers in cells are the filling factor. By the filling factor, it is possible to assume the filled condition of each cell. When the filling factor of a cell is equal to 1, the cell is perfectly filled by resin. And, the filling factor of an empty cell is equal to 0. At the resin flow front, the filling factor is between 0 and 1.

The filling factors are updated by calculating the below advection equation:

$$\frac{\partial \mathrm{I}}{\partial \mathrm{t}} + \mathbf{u} \cdot \nabla \mathrm{I} = 0 \qquad (2.1)$$

, which I is a filling factor, t time.

B. CVFEM

CVFEM is one of representative methods for RTM process simulation. To be more concrete, CVFEM is used for the discretization of the differential equation to calculate the pressure in the mold. And, this method is often combined with the VOF method to calculate the flow front position.

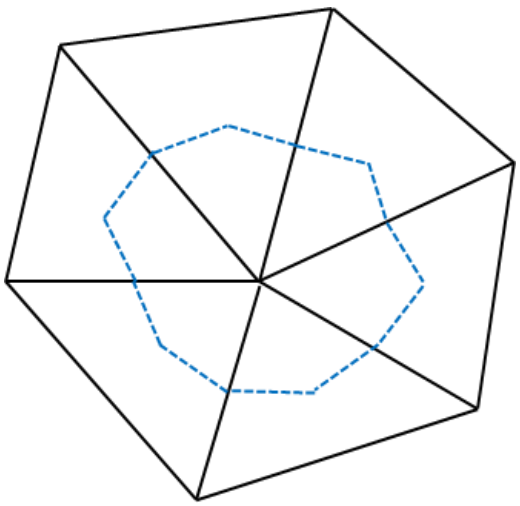


Figure 2.2. Control volume into three-node triangular elements

For applying CVFEM, control volumes should be defined into elements. An example of the control volume is illustrated in the figure 2.2. The control volume into the three-node triangular elements is discretized as the polygonal surrounded by dotted lines. The polygonal is made by the connection of centroids of the elements and midpoints of the corresponding sides. An integral formulation of differential equations by using CVFEM is obtained by applying the conservation principle for pressures or filling factors in the control volume [19,45,46].

C. Nonconforming FEM

Trochu et al. [24] proposed to apply the nonconforming FEM in numerical analysis of the RTM process. They approximated the pressure field by the nonconforming FEM. By contrast to the conforming FEM, the nonconforming FEM can locally satisfy the physical condition of resin mass conservation along the inter-element boundaries. To be concrete, the computed Darcy flow rates remain continuous across the boundary of elements. And, contrary to the CVFEM, the subdivision of the elements is unnecessary to account for the conservation of the resin mass. In the conforming FEM, the filling factors are nodal values. Contrariwise, in non-conforming FEM, the factors are based on the elements.

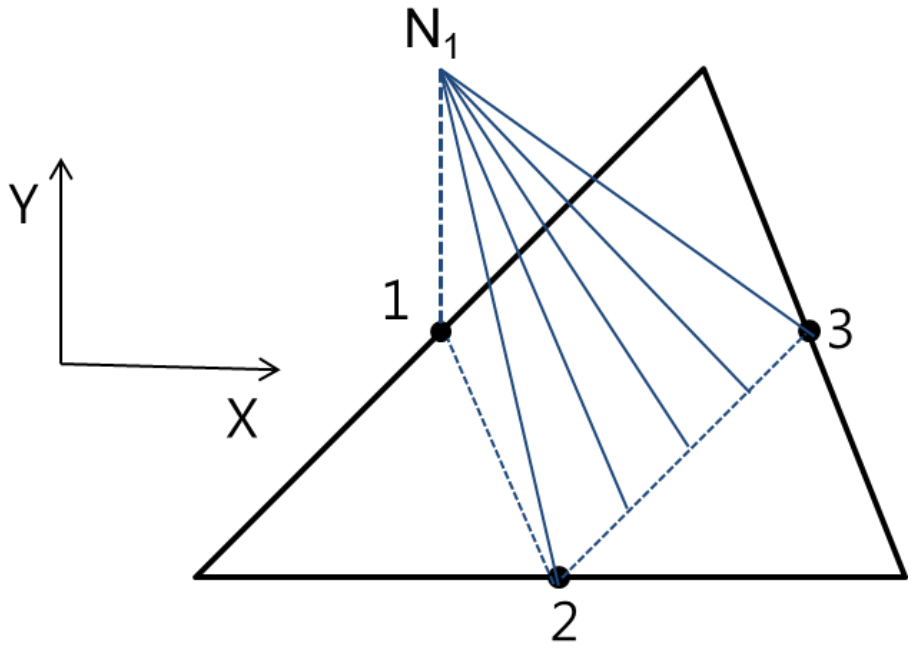


Figure 2.3. Shape function N_1 of a triangular element
for nonconforming FEM

The firuge 2.3 shows one of the shape functions on a nonconforming finite element. The pressure is interpolated by using linear shape functions N_i as the below equation [24]:

$$\mathrm{p(x,y)} = \mathrm{a} + \mathrm{bx} + \mathrm{cy} = \sum_i P_i N_i(x,y). \qquad (2.2)$$

And, the shape function N_i is defined as:

$$N_i(x_j^*, y_j^*) = \delta_{ij} = \begin{cases} 0, & if\ i \neq j \\ 1, & if\ i = j \end{cases} \qquad (2.3)$$

, where (x_j^*, y_j^*) are the coordinate of the middle nodes at the boundaries of elements [24].

2.2. XFEM and Level Set Method

A. Extended Finite Element Method (XFEM)

In this study, the pressure in the mold is calculated by using XFEM. An advantage of XFEM is to obtain accurately the pressure field without remeshing on the flow front region.

During mold filling in RTM process simulations, the derivative of the pressure function is weakly discontinuous at the resin flow front. The traditional fixed grid methods such as CVFEM and FEM do not reflect the discontinuity in the approximation. Despite the high computational efficiency of these fixed grid approaches, the continuously approximated pressure function at the flow front causes the low accuracy of simulation results.

XFEM is adopted so as to reflect the discontinuity of the pressure gradient field near the flow front. XFEM makes it possible to include the discontinuity in the interpolation function or its derivative. Therefore, XFEM approximation is more precise than traditional methods to calculate the pressure in the RTM process simulation.

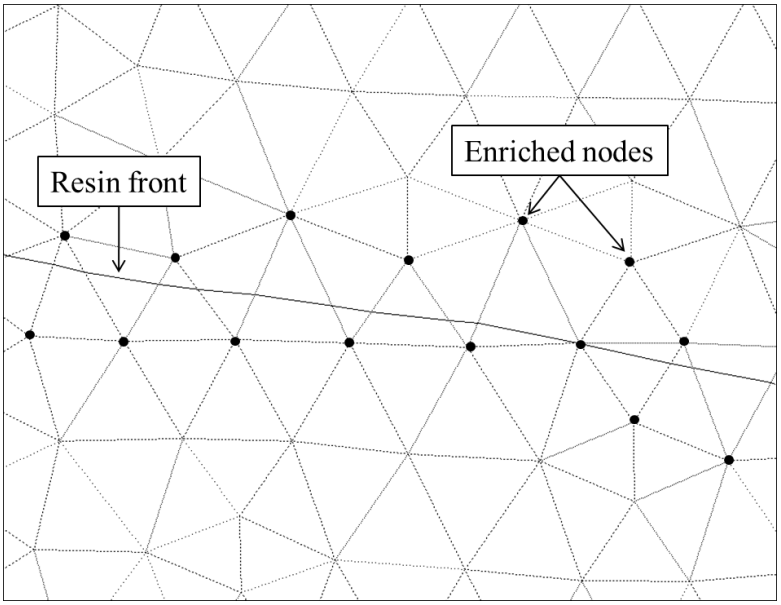


Figure 2.4. Definition of enriched nodes

The XFEM approximation of the pressure function is established by enrichment of the front region. The region for enrichment is defined shown in the figure 2.4. The enriched nodes are marked by dots in the figure 2.4. And, they are composed by the nodes of elements intersected by the flow front since the discontinuity exists at the flow front.

The enrichment means that additional degrees of freedom (DOF) are added in a classical finite element. The approximation of the pressure function for the enrichment is defined as:

$$p^h = \sum_{I \in n} N_I p_I + \sum_{J \in n_e} N_J^e a_J \qquad (2.4)$$

, where p^h is the pressure function, p_I the pressure value of the node I, a_J the additional nodal parameter at the enriched nodes J, N_I the shape

function, N_J^e the enriched shape function, n the number of nodes, and n_e the number of enriched nodes [57].

The enriched shape function is built as

$$N_J^e = N_J\left(|\varphi| - |\varphi_J|\right) \tag{2.5}$$

, where φ_J is a level set value of the enriched node J [57].

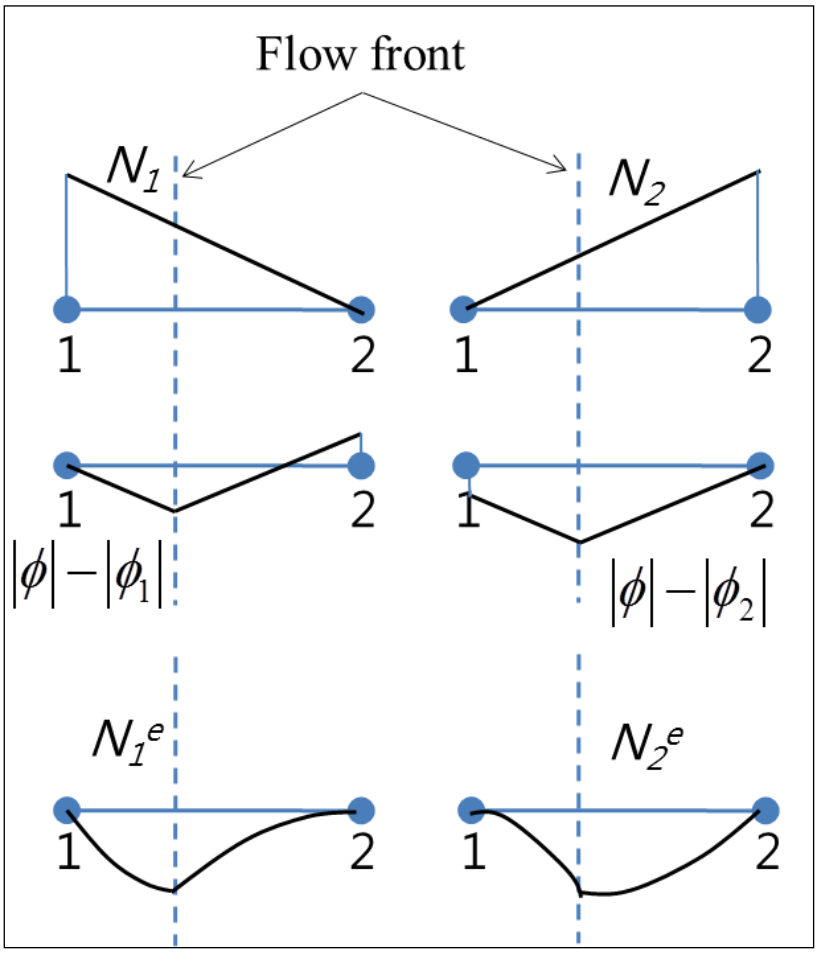


Figure 2.5. Enriched shape functions of one-dimensional linear element

The discontinuity at the flow front can be reflected by the functions consisting of the level set functions and values. For example, the enriched shape functions of a one-dimensional linear element are plotted in the figure 2.5. The significant character of the enriched shape functions is that the position of the flow front is reflected in the enriched shape functions. It is noted that the values of the enriched shape functions should be zero at the nodal points.

The pressure gradient at the flow front should be calculated carefully. The gradient of the approximated pressure function on the enriched region is obtained as [57]:

$$\nabla p^h = \sum_{I \in n} p_I \nabla N_I + \sum_{J \in n_e} a_J \left[\nabla N_J \left(|\varphi| - |\varphi_J| \right) + N_J sign(\varphi) \nabla \varphi \right]. \tag{2.6}$$

The jump in the approximated pressure function across the resin front is given as [57]:

$$[\![\nabla p^h]\!] = 2 \sum_{J \in n_e} N_J a_J \nabla \varphi \quad \text{on the resin front.} \tag{2.7}$$

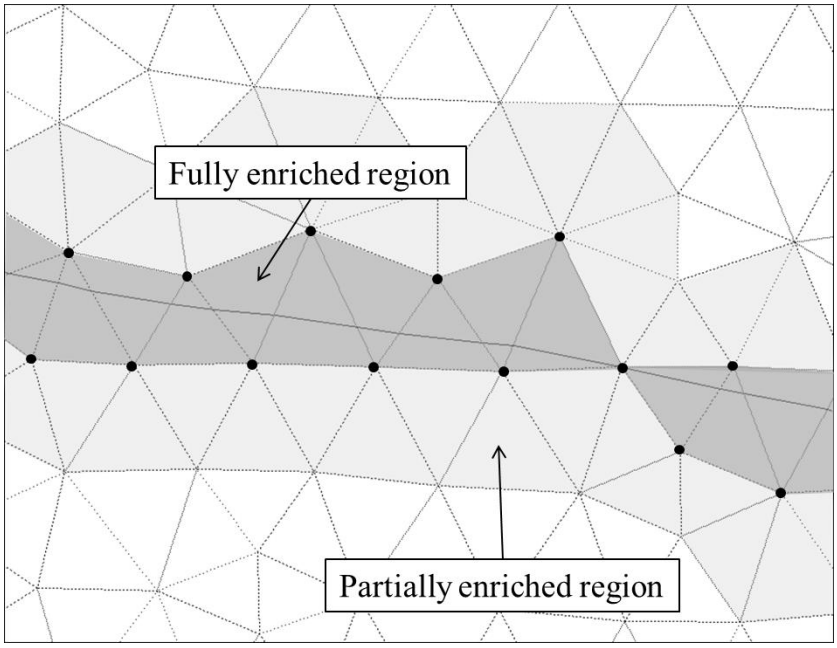


Figure 2.6. Types of enrichment

There are two types of enrichment as shown in the figure 2.6: full enrichment and partial one. The full enrichment concerns the elements having only enriched nodes. On the other hand, the partial enrichment concerns the elements having both enriched and normal nodes. The partial enrichment is necessary to keep the pressure between the fully enriched and normal regions continuous.

B. Level Set Method

In this study, the motion of the resin flow front is described by the level set method. The flow front position is defined by calculating the level set values.

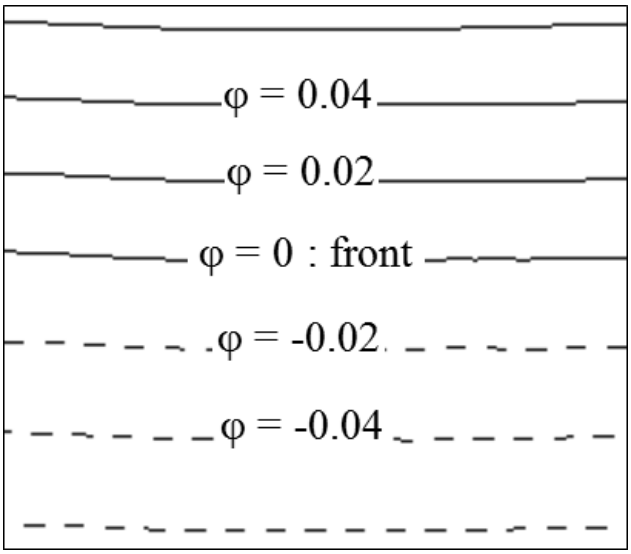


Figure 2.7. Level set values composed by the filled region, the front, and the unfilled region

The level set means the signed distance from the flow front as shown in the figure 2.7. The dashed lines are in the unfilled region, therefore, the level set values are negative signed distance from the front. The level set values at the flow front are equal to zero. In the filled region, marked by solid lines, the level set values are positive signed distance from the front.

The transport of the resin flow is established by solving a motion equation of the level set. The topological change in the flow

front is naturally handled by the motion equation of the level set. The motion equation is derived by the initial value formulation of J. A. Sethian [33]. The level set value of a particle on the front with path ***x(t)*** is equal to zero and the total derivative of the level set function becomes zero as:

$$\varphi(\boldsymbol{x}(t), t) = 0, \qquad \frac{D\varphi}{Dt} = 0. \tag{2.8}$$

By the chain rule, the differential equation (2.8) is derived as

$$\frac{\partial \varphi}{\partial t} + \mathbf{u} \cdot \nabla \varphi = 0. \tag{2.9}$$

Even though the velocity **u** is very changeable, the motion of the flow front can be naturally described by the equation (2.9).

However, the equation (2.9) can correctly transport just the flow front and cannot correctly change the level set values of the regions except the flow front.

In the level set method, a re-initialization procedure is necessary after solving the equation (2.9). The re-initialization procedure is to modify the level set values of the regions except the flow front to the exact distance values. The level set function should satisfy the condition as:

$$|\nabla \varphi| = 1. \tag{2.10}$$

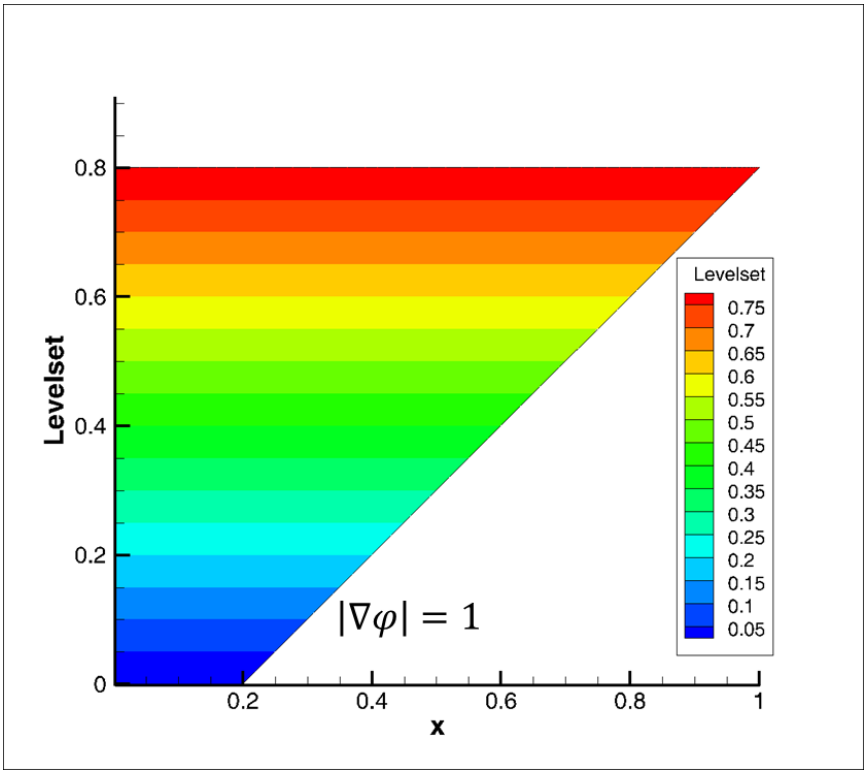


Figure 2.8. The ideal condition of a level set function

The figure 2.8 schematically shows the meaning of the equation (2.10). That is to say, the gradient value of the level set function is equal to 1.

Following the methodology of Ref. [31], the differential equation for re-initialization of the level set values are defined as:

$$\frac{\partial d}{\partial \tau} + \text{sign}(d_0)(|\nabla \text{d}| - 1) = 0 \qquad (2.11)$$

, where d is the signed distance function from the front, τ the imaginary time within the re-initialization procedure, and $sign(d_0)$ the sign function of the initial value d_0.

Additionally, the $sign(d)$ function is defined as

$$\text{sign}(d) = \frac{d}{\sqrt{d^2 + h^2}} \tag{2.12}$$

, where h is the mesh size [31].

The initial value of *d* is defined as [31]:

$$d(\boldsymbol{x}, 0) = d_0(\boldsymbol{x}) = \varphi(\boldsymbol{x}, t). \tag{2.13}$$

After the calculation, the level set function φ is re-initialized as

$$\varphi(\boldsymbol{x}, t) = d(\boldsymbol{x}, \tau). \tag{2.14}$$

The re-initialization procedure is repeated for several times at each time step of the equation (2.9).

Chapter 3. Analysis of RTM Process

This chapter delineates full details about analysis of RTM Process using XFEM and level set method. The analysis is composed into two major parts: the pressure calculation in mold and the transport of resin flow front. At first, the resin flow during mold filling is mathematically modeled. In addition, the formulations of the two parts are enumerated in detail. The total computational procedures are explained with a diagram. Finally, this study is validated with some simple models.

3.1. Modeling

A. Assumptions

For the analysis of mold filling stage, the resin flow in the mold is simplified with assumptions. The resin is assumed as an incompressible fluid. And, the mold filling stage is regarded as an isothermal condition since the mold filling time of the target models is shorter than the curing time. With this assumption, the resin viscosity could be constant during the mold filling. The permeability of the fiber preform in the mold could be constant due to the assumption that the fiber preform is considered as uniform and non-deformable during the mold filling.

The simplification by these assumptions has been used to set the model for the mold filling stage by many researchers [16, 17, 21, 24, 26, 29, 32, 43, 44, 71]. They simplified the resin flow in the mold as a quasi-steady state problem. The resin slowly flows; therefore, the inertia effect is insignificant. When the inertia effect is negligible, the resin flow could be assumed as a succession of a steady state flow. More specifically, the pressure in the mold is calculated at any instant of time t as a steady-state problem. After obtaining the pressure field, the fluid velocity is calculated. The resin flow front is transported by the velocity. In other words, the mold filling stage is mathematically modeled into two governing equations: for pressure calculation in the mold and transport of resin flow front.

B. Pressure Calculation

The figure 3.1 describes a domain for analysis of RTM process. The domain (Ω) indicates the phenomenon, which the dry preform is impregnated with resin entered through injection gates, and the air in the dry preform is ejected through vents.

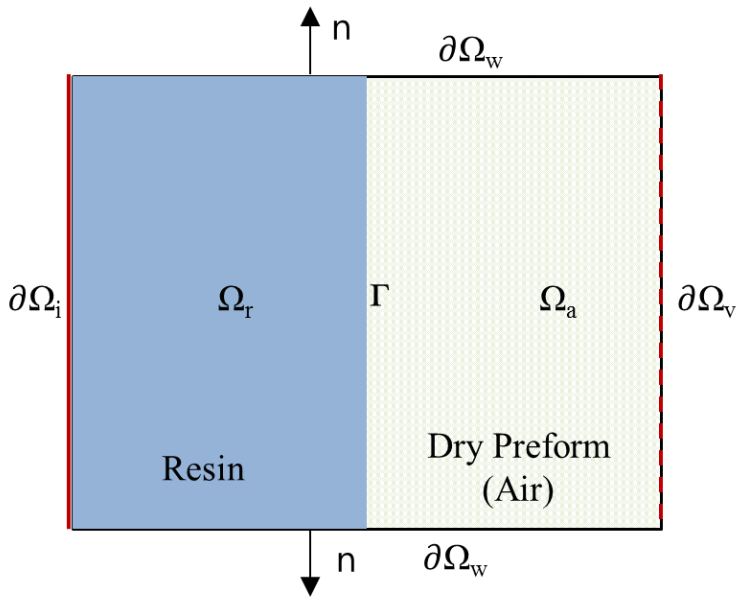


Figure 3.1. Computational domain for analysis of RTM process

In the figure 3.1, Ω_r is a wet region by resin. And, Ω_a is a dry region filled by air. Γ means a resin flow front, which is interface between resin and air. $\partial\Omega_i$, $\partial\Omega_v$, and $\partial\Omega_w$ are an injection port, vent, and mold wall, respectively.

During the mold filling, the resin and air in the mold are considered to be incompressible and isothermal fluids. Therefore, the mass conservation equation is given as [71]:

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega. \tag{3.1}$$

And, the momentum equilibrium equation is given as [71]:

$$\rho \, \mathrm{D}\mathbf{u}/\mathrm{D}t = -\nabla p - (\mu/\mathbf{K})\mathbf{u} \quad \text{in } \Omega \tag{3.2}$$

, where ρ is the density, t the time, p the fluid pressure, **K** the permeability tensor, and μ the fluid viscosity. The permeability tensor (**K**) and fluid viscosity (μ) is constant during the mold filling. In addition, because the inertia effect is neglected, the time derivative term is vanished in the equation (3.2). Therefore, the equation (3.2) is simplified as Darcy's equation in the steady-state flow as [71]:

$$\mathbf{u} = -(\mathbf{K}/\mu)\nabla p \quad \text{in } \Omega. \tag{3.3}$$

By substituting the equation (3.3) into the equation (3.1), the governing equation for the pressure field becomes as:

$$\nabla \cdot (-(\mathbf{K}/\mu)\nabla p) = 0 \quad \text{in } \Omega. \tag{3.4}$$

The boundary conditions are defined as the below equation (3.5). At injection gates, resin is injected by constant pressure or flow rate in the mold. Thus, the boundary conditions are set to injection pressure at injection gates as:

$$p = p_{injection} \quad \text{in } \partial\Omega_{\text{in}}. \tag{3.5 – a}$$

In addition, the pressure at vents is same with the air pressure as:

$$p = p_{air} \quad \text{in } \partial\Omega_{\text{v}}. \tag{3.5 – b}$$

At mold walls, the resin flow cannot pass through mold walls; therefore, the partial derivative of pressure with respect to the normal direction of mold walls is equal to zero as:

$$\frac{\partial p}{\partial n} = 0 \qquad \text{in } \partial\Omega_w. \tag{3.5 – c}$$

Then, the pressure field and fluid velocity in the mold are solved at each time as steady-state problem, the resin flow front is transported for the next time step.

B. Transport of Flow Front

The transport of the resin flow front is transported by solving the following equation:

$$\partial\varphi/\partial t+\mathbf{u}\cdot\nabla\varphi=0 \qquad \text{in } \Omega \tag{3.6}$$

, where t is the time, φ the level set function, and **u** the fluid velocity [33]. By solving the equation (3.6), the level set values are updated at each time step. The details to derive the equation (3.6) are explained in Section 2.2. The initial condition of the level set values is the distance from the injection gates.

To correctly transport the flow front, re-initialization of level set values should be performed after solving the equation (3.6). The details for the re-initialization are presented in Section 2.2.

3.2. Formulation

A. Pressure Calculation

The differential equation for the pressure calculation is solved by using XFEM. The XFEM formulation is similar with the general FEM. The weak form of the differential equation for the pressure is obtained as:

$$\int_{\Omega}(\nabla\cdot\left(-\frac{\mathbf{K}}{\mu}\nabla p\right))wdV=0 \qquad (3.7)$$

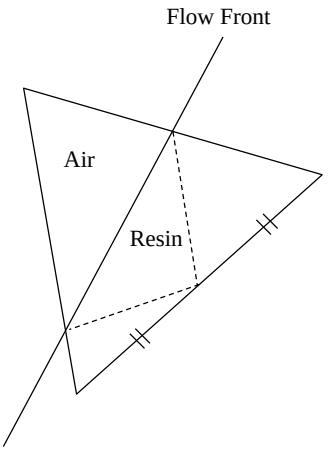
, where *w* is the weight function. Applying the Green-Gauss theorem, the equation (3.7) is transformed as:

$$\int_{\Omega}\nabla \mathrm{w}\cdot\left(-\frac{\mathbf{K}}{\mu}\nabla p\right)dV-\int_{\partial\Omega}\mathrm{w}\left(-\frac{\mathbf{K}}{\mu}\nabla p\right)\cdot\mathbf{n}dS=0 \qquad (3.8)$$

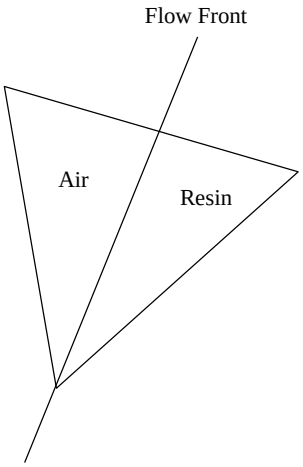
, where **n** is the outward normal vector. By XFEM, the pressure and weight functions are discretized as the equation (2.4). In other words, the approximation of the functions is done by adding the additional terms for the enrichment in the terms of the standard Galerkin FEM.

The integral of the equation (3.8) is calculated by the Gauss quadrature method. The elements intersected by the flow front are divided into two parts: filled and unfilled regions. These two regions

are separately integrated because the derivative of the pressure function is weakly discontinuous at the flow front. At the normal region, the integration performed by the Gauss quadrature method in general FEM without any special treatment. However, for the integral of fully enriched regions, subdivision of fully enriched elements should be performed before the quadrature method.



(a) Case 1



(b) Case 2

Figure 3.2. Subdivision in elements intersected by flow front

For the integral at the fully enriched region, the elements intersected by the flow front are divided into several parts. As shown in the figure 3.2, four or two sub-domains are constructed. When flow front passes through two sides of a triangle, the element is divided by four sub-domains as shown in the figure 3.2 (a). In addition, when flow front passes through one point and one side of a triangle, the element is divided by two sub-domains as shown in the figure 3.3. (b). The integral of each sub-domain is individually calculated by the Gauss quadrature rule.

The permeability **K** of fiber preform is identical in the whole domain. However, the viscosity μ of resin is different at each region. At the filled region by resin, the resin viscosity is applied. And, the empty region has air viscosity. The viscosity of the region intersected by resin flow front should be an averaged value of resin and air values Therefore, the fluid viscosity is defined as:

$$\mu = H(\varphi)\mu_r + (1 - H(\varphi))\mu_a \qquad (3.9)$$

, where μ_γ is the resin viscosity; μ_α the air viscosity, and ϕ the level set value. Here, $H(\varphi)$ is defined as

$$H(\varphi) = \begin{cases} 1 & if\ \ \varphi < -h \\ \frac{1}{2}\left(1 - \frac{\varphi}{h}\right) & if\ \ |\varphi| \le h \\ 0 & if\ \ \varphi > h \end{cases} \qquad (3.10)$$

, where h is the mesh size.

B. Transport of Flow Front

The differential equations for transport of flow front are solved by the implicit characteristic Galerkin FEM [30]. The implicit characteristic Galerkin FEM is employed to stabilize the non-self adjoint term. The time derivative term is discretized by the implicit characteristic method as [65]:

$$\varphi^{n+1} - \varphi^{n} = -\Delta t[\mathbf{u} \cdot \nabla\varphi]_{n} + \frac{\Delta t^{2}}{2}\left[\nabla \cdot \left((\mathbf{u}\otimes\mathbf{u}) \cdot \nabla\varphi\right)\right]_{n+1} \tag{3.11}$$

, where Δt is the size of the time step. For the discretization of the spatial differential terms in the equation (3.11), the standard Galerkin FEM is applied as:

$$\int_{\Omega}(\varphi^{n+1} - \varphi^{n})w dV = -\Delta t\int_{\Omega}(\mathbf{u} \cdot \nabla\varphi^{n})w dV - \frac{\Delta t^{2}}{2}\int_{\Omega}\left((\mathbf{u}\otimes\mathbf{u}) \cdot \nabla\varphi^{n+1}\right) \cdot \nabla w dV \tag{3.12}$$

, where *w* is the weight function and Ω is the whole space in the mold. The velocity **u** for transport of the front is calculated by the equation (3.3).

The re-initialization equation is also discretized by the procedures of the equation (3.11) and (3.12) as:

$$\int_{\Omega}(d^{n+1}-d^{n})wdV = -\Delta\tau\int_{\Omega} sign(d_0)|\nabla d^{n}|wdV - \frac{\Delta\tau^{2}}{2}\int_{\Omega}\nabla d^{n+1}\cdot\nabla wdV + \Delta\tau\int_{\Omega} sign(d_0)\mathrm{wdV} \tag{3.13}$$

, where $\Delta\tau$ is the size of the time step in the re-initialization procedure.

In the equation (3.13), the second term of the right-hand side could be derived by the next procedure:

$$sign(d_0)|\nabla d| = sign(d_0)\frac{\nabla d}{|\nabla d|}\cdot\nabla d\,, \qquad \left(sign(d_0)\frac{\nabla d}{|\nabla d|}\right) sign(d_0)|\nabla d| = \nabla d. \tag{3.14}$$

The coefficient matrices of systems obtained by the formulations are symmetric and positive definite; therefore, the multi-frontal solver of IPSAP can be applied.

3.3. Computing Procedures

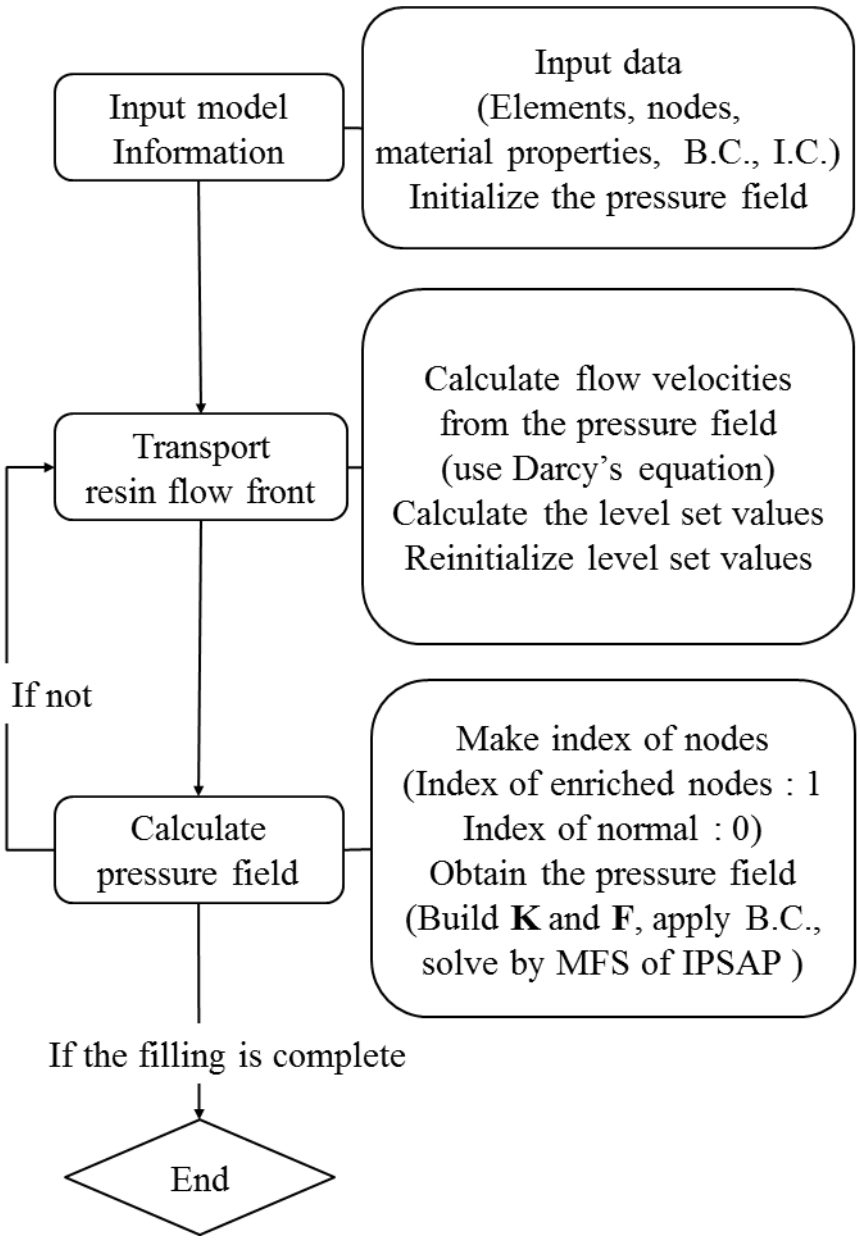


Figure 3.3. Flow chart of the computational procedures

The whole computation scheme is shown in the figure 3.3. The details are explained as follows. The first step is to read the input data, such as the mesh information, material properties, and boundary conditions, and to initialize the pressure field and level set values. At the second step, the fluid velocity is calculated from the pressure field by using the Darcy's equation, explained in Section 3.1. The level set values are calculated in order to transport the resin front. The re-initialization of the level set values is performed at each time step. Then, the index of the nodes is set by the level set distribution: enriched nodes by 1 and normal nodes by 0 to denote the corresponding shape functions. The second and third steps are continued until the mold filling is complete. The multi-frontal solver of IPSAP [63, 64] was applied to solve the discretized equilibrium systems. The multi-frontal solver of IPSAP will be introduced in Appendix A.

3.4.Validation

3.4.1. Pressure Calculation

The pressure calculation by XFEM is verified by the comparison with analysis results of COMSOL Multiphysics, which is commercial software for engineering simulation.

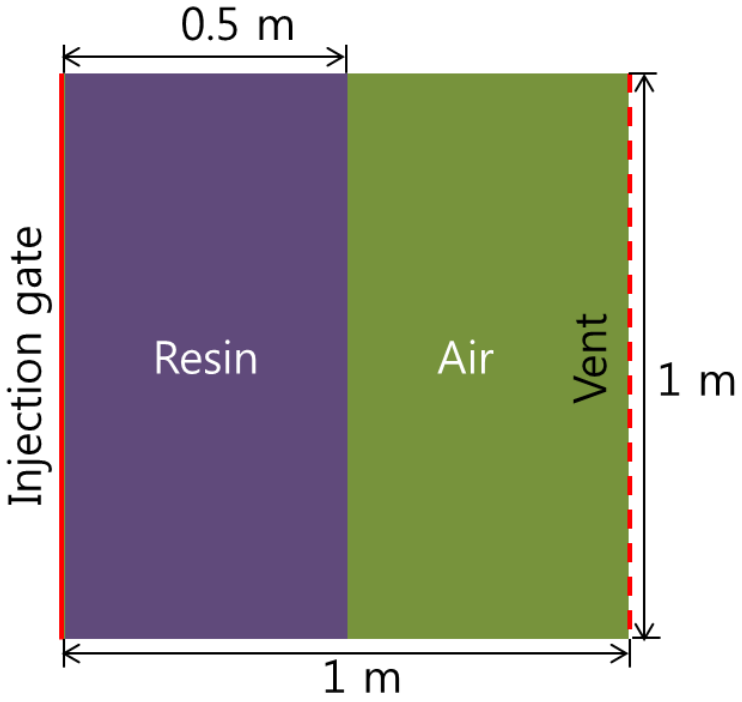


Figure 3.4. A simple square plate

A simple square plate, which has the width of 1m, the height of 1m, and the thickness of 1mm, is taken for the verification as shown in the figure 3.4. The injection gate and vent are place at the left side (a red solid line) and light side (a red dotted line), respectively. When the

resin flow front is reached at the 0.5 m point from the injection gate, the pressure in the mold is calculated by XFEM and COMSOL Physics. For the material properties, the viscosity values of resin and air are set to 0.1 Pa·s and 0.00001 Pa·s, respectively. And, the permeability is $10 \times E^{-8}$. The element type of both models is three-node triangle.

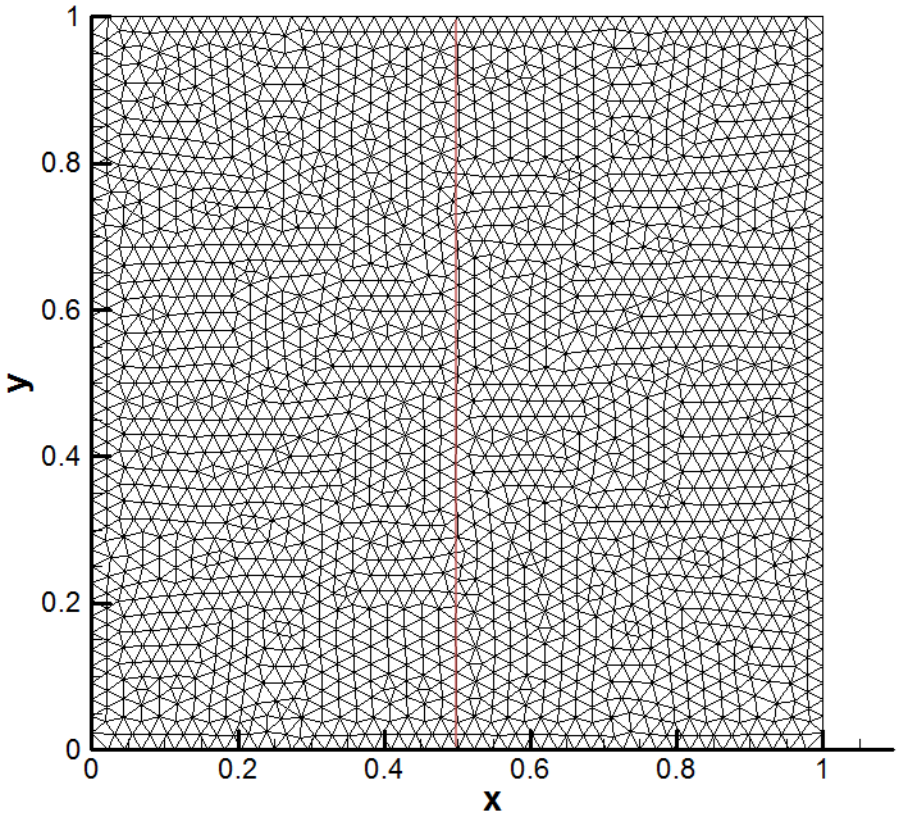


Figure 3.5. Mesh of a simple square plate for XFEM

To exactly check the quality of XFEM, the mesh shape is unstructured. The mesh conditions for the two programs are different. For using XFEM, the mesh of the model is set as the figure 3.5. The red line at the middle is placed on the flow front region. As shown in the figure 3.5, the elements near the flow front are intersected by the red line (flow front).

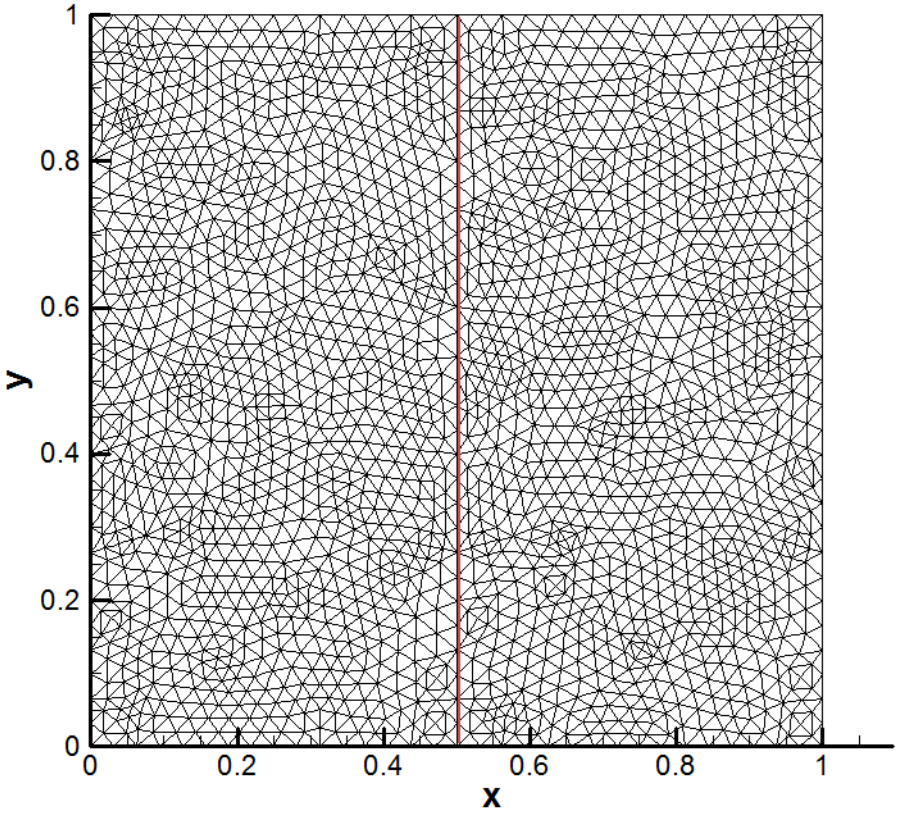
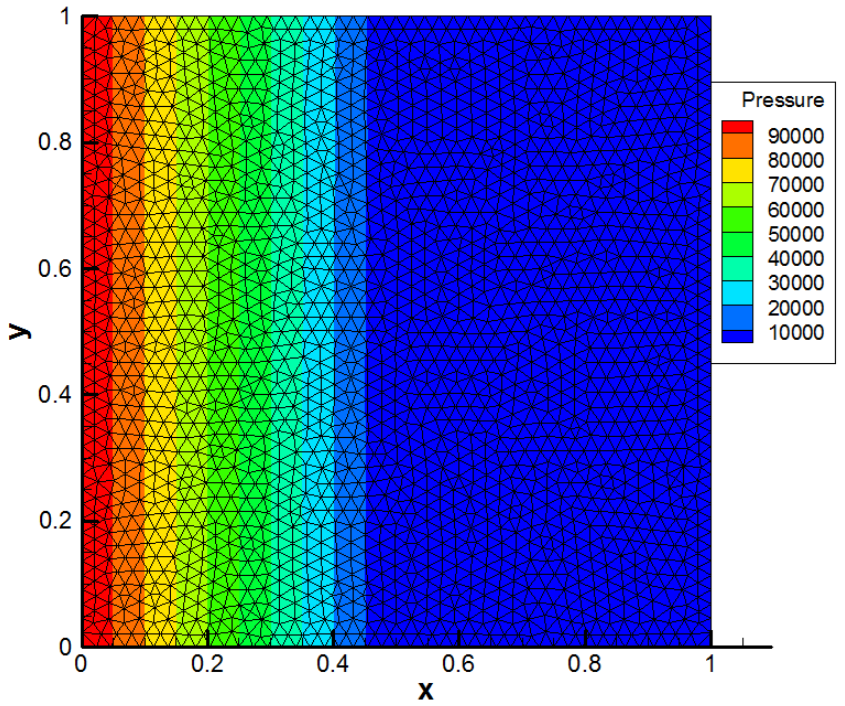
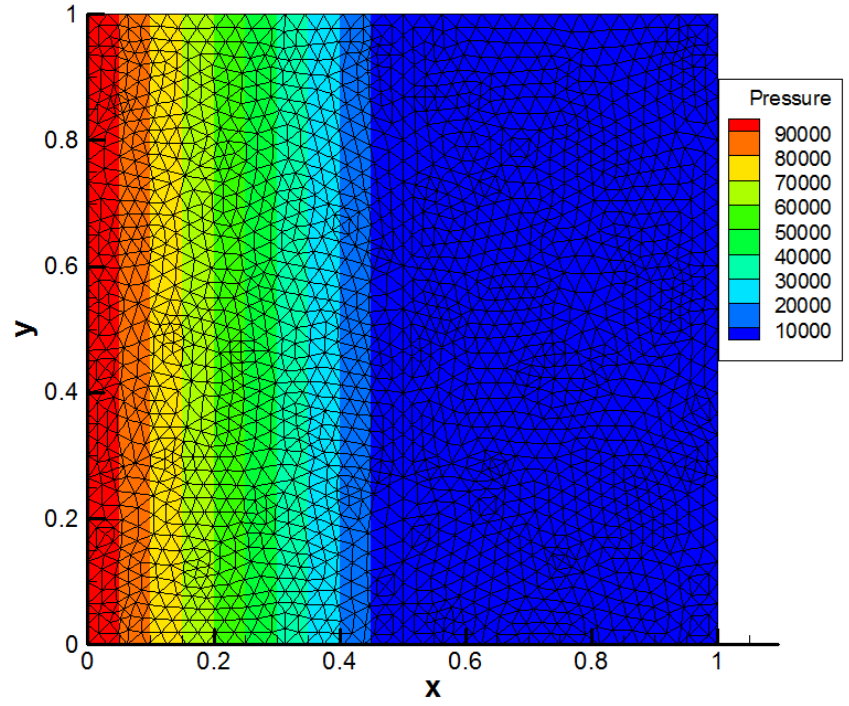


Figure 3.6. Mesh of a simple square plate
for COMSOL Multiphysics

COMSOL Multiphysics is applied FEM for the simulation and offers the mode for solving the general partial differential equation (PDE). The PDE mode is used to calculate the pressure of the simple square plate. For applying FEM, the meshed condition of the model is preserving the flow front line (red line) as shown in the figure 3.6. The number of elements and nodes are 2043 and 3960, respectively.



(a) By XFEM



(b) By COMSOL Multiphysics

Figure 3.7. Pressure contours of a simple square plate

The analysis results are plotted in the figure 3.7. The figure 3.7 (a) is the pressure contour obtained by using XFEM. And, the figure 3.7 (b) is the one calculated by COMSOL Multiphysics. As shown in the figure 3.7 (a) and (b), the two pressure contours are almost similar. Even when the mesh shape of model is unstructured and the resin flow front intersects the elements, the XFEM well works to calculate the pressure field.

3.4.2. Transport of Flow Front

The part for transport of flow front is verified through simple models.

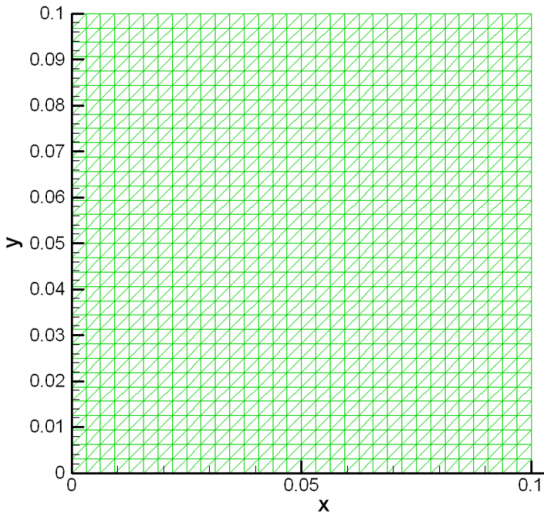


Figure 3.8. Geometry and mesh information of a thin square plate for verifying level set calculation

The geometry and mesh information of the models is shown in the figure 3.8. The model is a thin square plate which has the width and height of 0.1 m. And the thickness is 1 mm. The type of element is three-node triangle. The number of elements and nodes are 2048 and 1089, respectively.

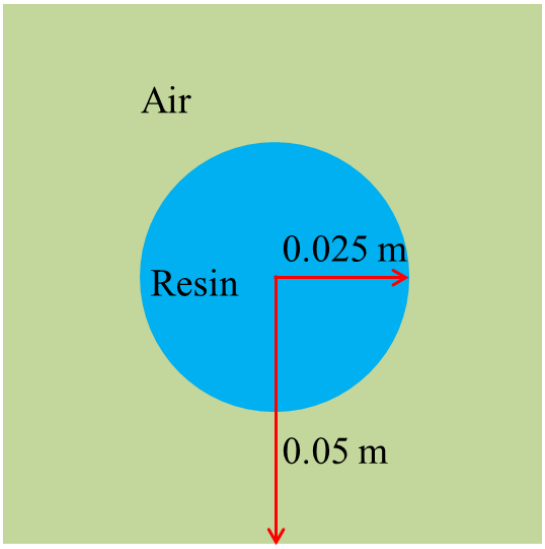


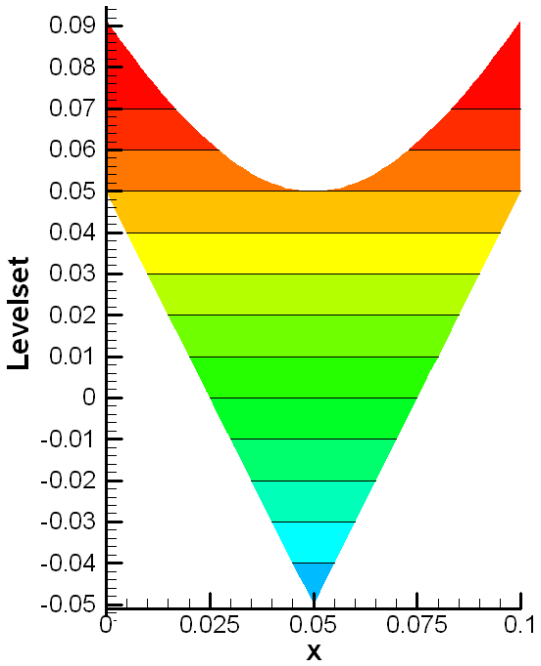
Figure 3.9. A model for the verification of the re-initialization procedure

At first, the verification of re-initialization procedure is performed by using the model described in the figure 3.9. The model is filled with resin until the circle region which is 0.025 meters in radius. The circle region is placed at the middle of the square. The region except the one surrounded by the circle is filled with air.

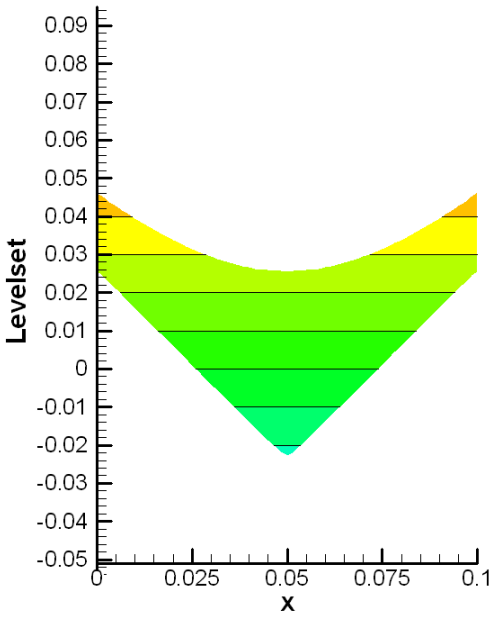
The initial condition of level set is given as:

$$|\nabla\varphi| = |\nabla d_0| = 2. \qquad (3.15)$$

This condition means the gradient of level set values is 2. After the re-initialization computation, the level set values should be changed from the condition of the equation (3.15) to the correct condition: $|\nabla\varphi| = 1$.



(a) Initial step



(b) 70th step

Figure 3.10. Change of level set values by the re-initialization procedure

The change of level set by re-initialization calculation is shown in the figure 3.10. As shown in the figure 3.10 (a), at the initial step, the gradient of level set function is equal to 2. After seventieth time step, the gradient of level set function becomes 1 as shown in the figure 3.10 (b).

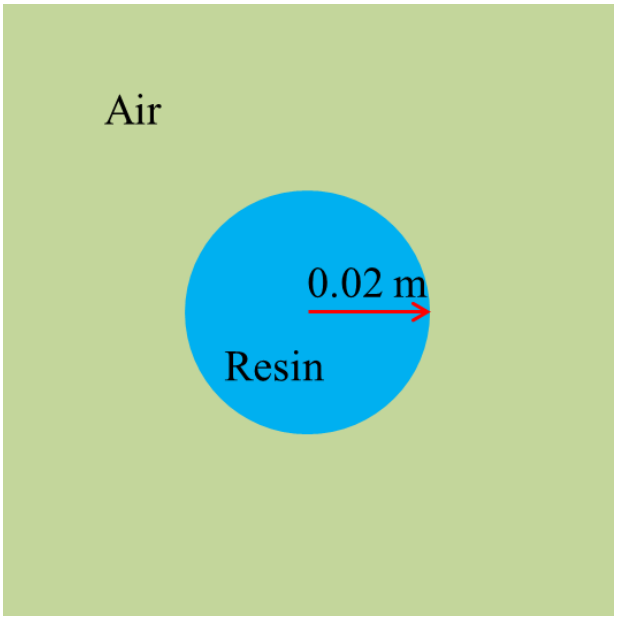
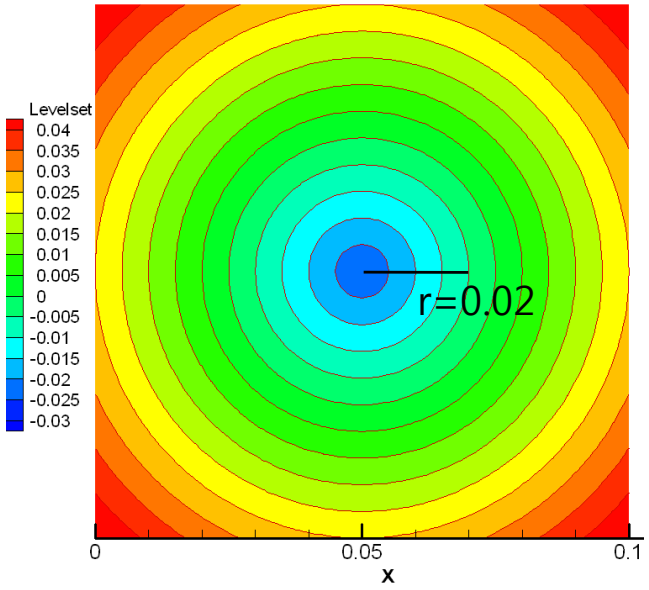
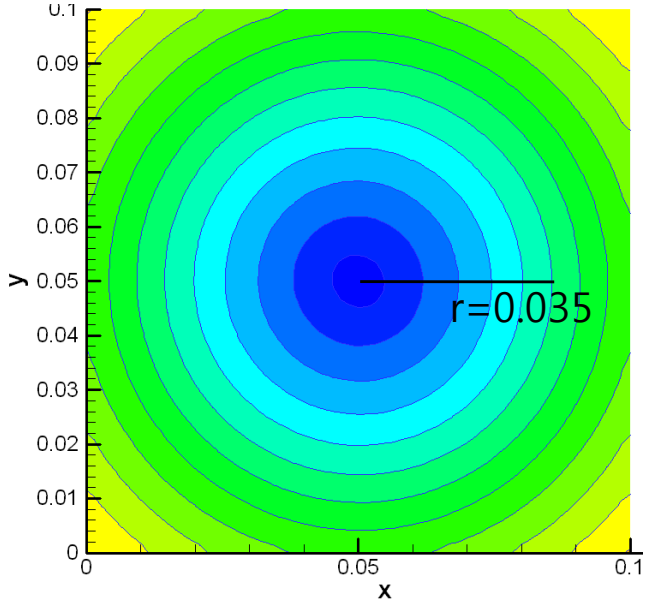


Figure 3.11. Thin square plate with the uniformly expanded fluid flow

The total level set calculation procedure including the transport of flow front and the re-initialization is verified by a uniform expansion flow model. The model is shown in the figure 3.11. At the initial condition, the resin front is placed at the region of 0.02 meters in radius from the center. The velocity of fluid flow is uniform and the flow is expanded to the normal direction of flow front.



(a) Initial step



(b) 20th step

Figure 3.12. Change of level set values by the level set calculation procedure including the transport of flow front and the re-initialization

The figure 3.12 shows the change of level set values after the calculation. At the initial step, the flow front is placed on the 0.02 m in

radius from the center. After twentieth steps, the flow front is uniformly expanded until 0.025 m in radius.

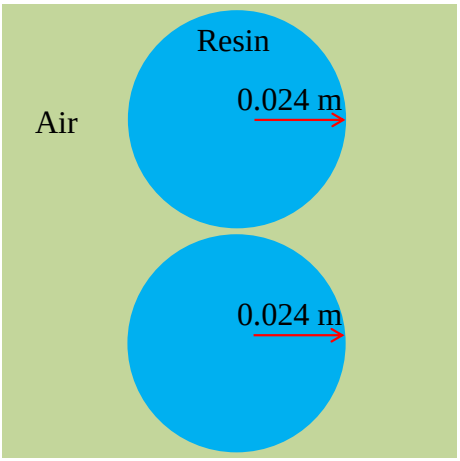
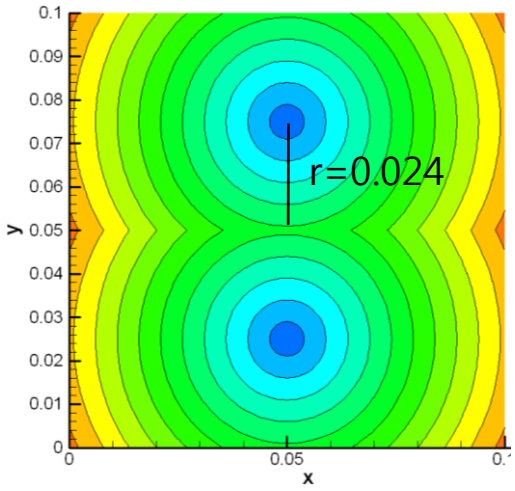
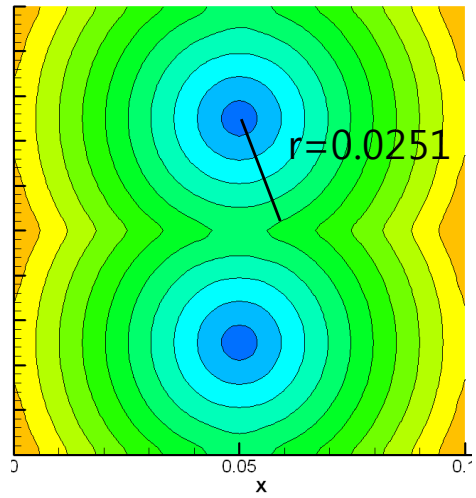


Figure 3.13. Plate for uniform expansion flow with two fronts

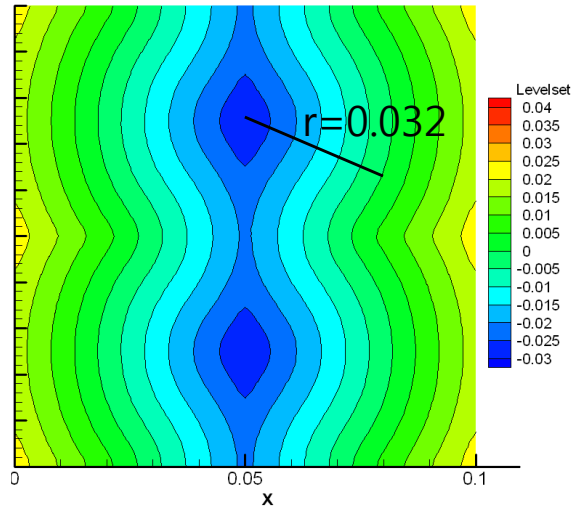
For testing the merging of flow fronts, thin square plate with uniformly expanded fluid flow at two flow fronts is used. Two resin fronts are placed on the two points as shown in the figure 3.13. The fluid flow also has constantly expanded velocity field.



(a) Initial step



(b) 1st step



(c) 10th step

Figure 3.14. Merging of two flow fronts

The results are shown in the figure 3.14. At initial step, the two fronts do not touch each other. The two fronts touch each other from the first step. After tenth step, the two fronts are perfectly merged.

3.4.3. Comparison with Analytic Result

Simple channel or radial flow examples are treated to validate the present methods in comparison with analytical solutions. The total filling time obtained in this study is compared with the analytical one proposed by Cai [66]. A three node triangular element is used in this analysis

A. 2D Channel Flow in a Rectangular Plate

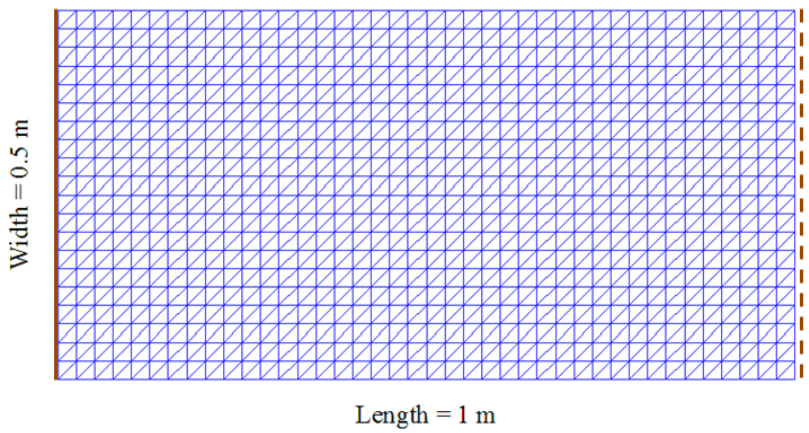


Figure 3.15. Geometry and mesh of the channel flow rectangular plate

The figure 3.15 shows a 2D channel flow of rectangular plate with a constant width. Its length, width and thickness are equal to 1 m, 0.5 m and 1 mm, respectively. The resin is injected at the whole left

side of the plate, as marked by the solid line. And the resin goes out from the mold through the whole right side of the plate (dashed line). The numbers of total nodes and elements are 861 and 1600, respectively. Material properties and boundary conditions of the model are set as follows: resin viscosity = 0.1 Pa·s, preform permeability (isotropic and constant) = 10^{-9} m^2, porosity = 0.5, injection pressure = 20 MPa, vent pressure = 0.

The analytic mold filling time for the channel flow, when the injection pressure is identical to the width direction, is given as:

$$t_{filling} = \frac{L^2 \mu \phi}{2KP} \tag{3.16}$$

where L means the position of the flow front in the length direction. And, μ, ϕ, K, and P are the viscosity, porosity, permeability, and pressure, respectively.

The analytical total filling time by the equation (3.16) is equal to 1250 seconds. The numerical total filling time obtained in this study is equal to 1268.42 seconds. The difference between the analytical and numerical results is less than 1.5%. In comparison with the numerical solution (difference of 3%) proposed by S. soukane and F. Trochu [32], which treated the same example, our program is more accurate.

B. 2D Radial Flow Example in a Circular Plate

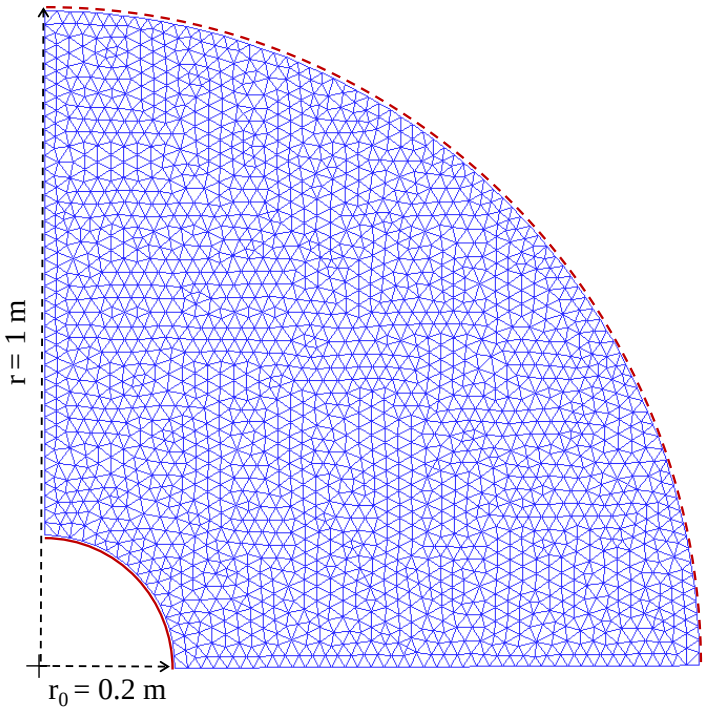


Figure 3.16. Geometry and mesh of the radial flow model

The figure 3.16 shows the radial flow model with a thin circular disk. Because of its axisymmetric conditions, a quarter of the disk is modeled. The radius of the disk is 1 m. And the thickness are equal to 1 mm. The resin is injected through a hole located at the center, of which the radius is equal to 0.2 m. The number of nodes and elements are 2082 and 3987, respectively. The dashed line at the outer side of the plate means the vent location. The value of resin viscosity, preform permeability, porosity, and injection pressure are the same as in the previous example (2D channel flow example).

The analytic mold filling time of the channel flow model is obtained by the following equation:

$$t_{filling} = \frac{\mu\phi\, r_0^{\,2}}{4KP}\left(\frac{r}{r_0}\right)^2\left(2\ln\frac{r}{r_0} + \frac{r_0^{\,2}}{r^2} - 1\right) \qquad (31)$$

, where r is the radius of the plate and r_0 the radius of the hole. The analytical filling time of the radial flow example is equal to 1411 seconds. The numerical filling time obtained in this study is 1436 seconds. The difference between the two results is 1.8 %. Our program is more accurate in comparison with Mark Lin et al. [72] which gives 4% of difference with the analytical solution.

Chapter 4. Efficiency of Localization

This study presents localization of XFEM and level-set method for the improvement of computational efficiency through minimization of the computational quantity. The localization means to allocate the small region near the flow front. In the problem to simulate the resin flow in the mold, the target is the description of the resin flow front movement. Therefore, the level set values except the region near the resin flow front are unnecessary. It is possible to increase the computational efficiency if the domain is reasonably reduced to obtain only the necessary level set values without any influence about the results. Generally, while solving partial differential equations, the most time-consuming procedure is when calculating the linear system. Therefore, if the system size (DOF number) becomes much smaller, the total computation time could be certainly reduced. Even though another computation is included for allocation of the domain, the total computation time might be appreciably decreased because the time for the allocation is relatively very less. In this chapter, the algorithm for the localization for XFEM and level-set method is introduced. Furthermore, the computational efficiency by the localization is investigated with some examples.

4.1. Level Set Method

This algorithm, which is used for localization of the level set method, is originated from the PDE-based fast local level set method algorithm proposed by Peng et al [31]. In this study, their algorithm is modified to apply in the implicit characteristic Galerkin FEM.

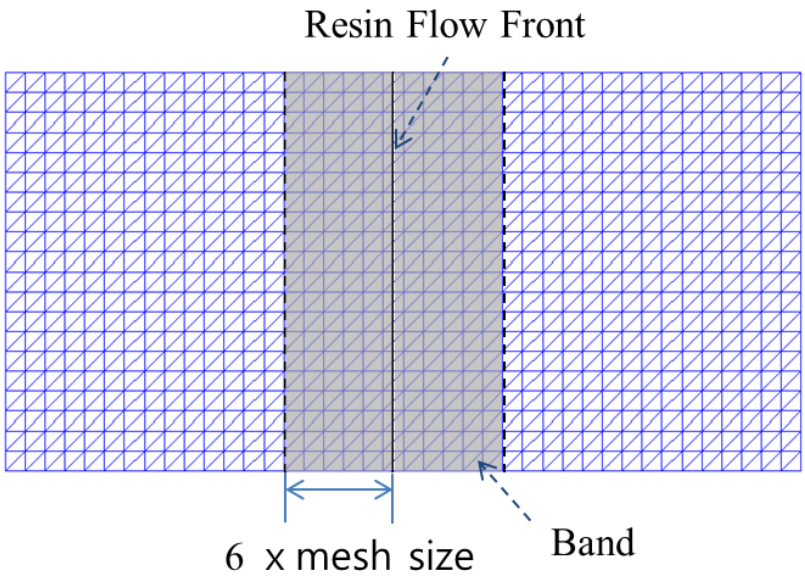


Figure 4.1. Computational domain for localization of the level set method

A region near the flow front is allocated for the localization of level set method as shown in the figure 4.1. The band marked by grey is the allocated region. The band is newly defined at each time step while moving the position of the flow fronts. A distance from the resin flow front to the band boundary is six times as long as the mesh size. The level set values of nodes are updated only in the band at each time step. Therefore, the computation amount becomes very less.

Peng et al.[31] proposed the band formation algorithm to apply in the FDM. In this study, their algorithm is modified in order to properly work the algorithm in the FEM. The differences with their algorithm are change of the procedure to define the list of nodes for reinitialization and addition of the procedure to define the list of elements in the band.

```
gamma=6*size_mesh;
for(i=0;i<number_node;i++){
    mask[i]=0;
    if(fabs(levelset[i])<gamma){
        mask[i]=2;
        list[num_local_node]=i;
        list_re[num_local_node_re]=i;
        num_local_node++;
        num_local_node_re++;
    }
    else if(fabs(levelset[i])<(gamma+mesh_size)){
        mask[i]=1;
        list_re[num_local_node_re]=i;
        num_local_node_re++;
    }
    else{
        list_left[num_left_node]=i;
        num_left_node++;
    }
}

for(i=0;i<num_element;i++){
    for(j=0;j<num_local_node;j++){
        if(mask[n[j]]==2)band_node++;
        else if(mask[n[j]]==1)band_boundary_node++;
    }
    if(band_node==num_local_node){
        list_element[num_local_element]=i;
        num_local_element++;
    }
    if((band_node+band_boundary_node)==num_local_node){
        list_element_re[num_local_element_re]=i;
        num_local_element_re++;
    }
}
```

Figure 4.2. Pseudo-code of the algorithm to define the band region

The figure 4.2 is the pseudo-code of the algorithm to define the lists of nodes for the band formation in this study. There are three lists of nodes. The first list (“list”) includes the nodes in the band. And, the nodes in the band and its boundary are collected in the second (“list_re”), which is used for the reinitialization. The nodes not included in the first and second lists are saved at the third list (“list_left”).

To explain the pseudo code, at first, the band size is chosen. Next, the level set values of every node are scanned to judge whether the node is in the band or outside of the band. By comparing the level set values of the nodes with the band size, the nodes are indexed by the number 0, 1, or 2, and classified into the three lists. The index of nodes is used while the elements are assorted. The code makes the lists of elements through scanning the index of the nodes of each element. The first list is for the element in the band. And, the second is for the elements in the band and its boundary. Only the elements included in the two lists are called when constructing the equilibrium system to calculate the level set values at each time step.

For the localization of level set method, a cut-off function is necessary to prevent numerical oscillations at the band boundary. The cut-off function is defined as:

$$c(\varphi) = \begin{cases} 1 & if\ |\varphi| \le \beta \\ (|\varphi| - \gamma)^2(2|\varphi| + \gamma - 3\beta)/(\gamma - \beta)^3 & if\ \beta < |\varphi| \le \gamma \\ 0 & if\ |\varphi| > \gamma \end{cases} \quad (4.1)$$

, where φ is the level set value, γ six times the mesh size, β three times the mesh size [31]. By applying the cut-off function in the differential equation to calculate the level set values, the governing equation is modified as:

$$\frac{\partial \varphi}{\partial t} + c(\varphi)\mathbf{u} \cdot \nabla \varphi = 0. \qquad (4.2)$$

The level set values, obtained by the equation (4.2), are correct in the space within β distance from the resin flow front. In the space between γ distance and β distance from the front, the values are modified by the cut-off function. Outside of the band of γ distance from the resin flow front, they are not updated and simply initialized to γ at the last.

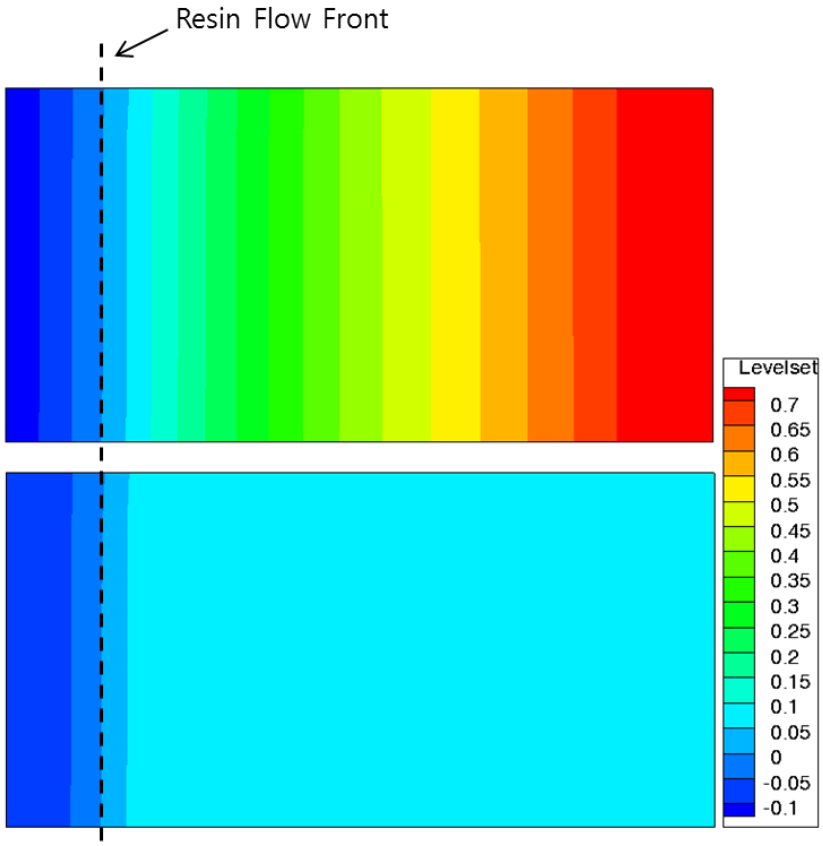


Figure 4.3. Level set contours obtained without (Top) and with (Bottom) localization of the level set method

A channel flow example is used to show the difference of the two calculations without and with the localization of the level set method. The example information for modeling is explained in Section 3.4.3. The figure 4.3 shows the contour of level set values at the early stage of the mold filling. As shown in the top contour of the figure 4.3, change of the values is indicated at the whole domain because they are obtained by calculation without the localization. However, the bottom contour shows the change only in the band. At the region out of the band, the values are set to 0.15 or -0.15 which is the distance of the band boundary from the front.

4.2. XFEM

The localization of XFEM is attempted to reduce the computational amount at the part of the pressure calculation. As aforementioned, the pressure in the mold is used to calculate the flow velocity. By applying the localization of level set method, the pressure outside of the local computation domain is not used in the level set calculation. This study makes the local computation domain for obtaining the pressure and avoids the unnecessary computation.

The key point for the localization of XFEM is how to apply the boundary conditions. Because the initial value formulation is used at the level set calculation, the boundary conditions are not considered for the localization of the level set method. However, the pressure is

calculated by the boundary value formulation; therefore, the boundaries changed at each time step might be considered in the localization of XFEM.

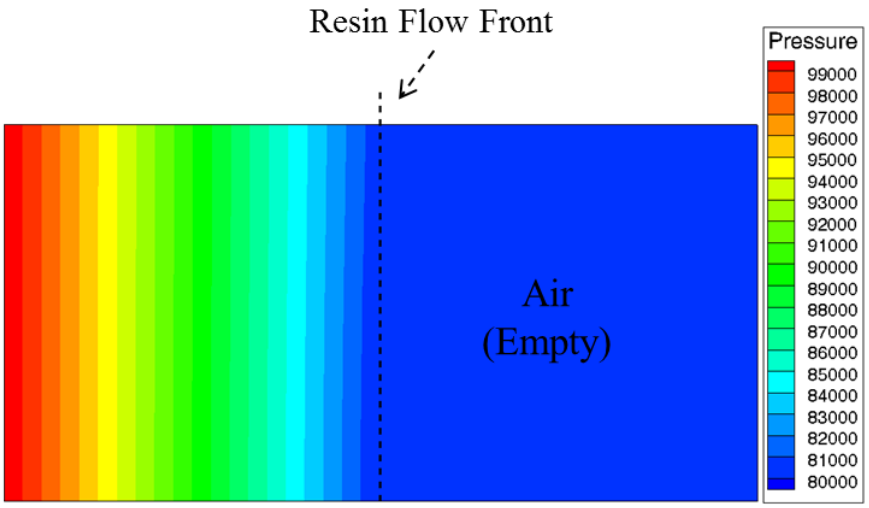


Figure 4.4. Pressure contour of the channel flow model

The figure 4.4 is a pressure contour in the mold. The pressure contour is obtained by the XFEM without localization. As shown in the figure 4.4, the air pressure in the mold is very less changed than the resin pressure. Even though the pressure in the empty region little change, the pressure of the air region in the band, which is the local domain for the level set calculation, is necessary to obtain the flow velocity. Therefore, this study calculates just the pressure in the necessary region.

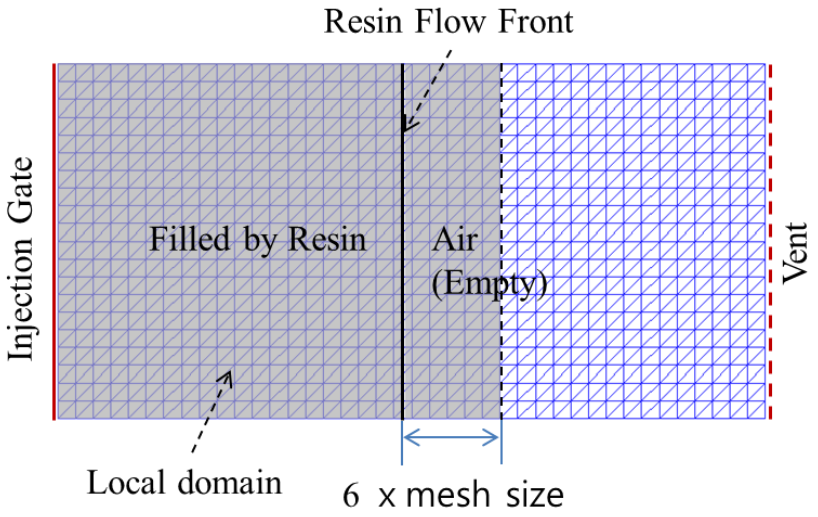


Figure 4.5. Localized computational domain (Localization of XFEM)

The local domain for the pressure calculation is defined as shown in the figure 4.5. The local domain includes the filled region by resin and the small portion of the air region. The portion is defined from the flow front to six times the mesh size.

The boundary of empty side in the local domain is set to the vent pressure because the change of the pressure in the air region is very less. The boundary conditions at the mold wall, injection gates, and vents are same with the equation (3.5).

```
gamma=6*size_mesh;
for(i=0;i<num_node;i++){
    mask[i]=0;
    if(levelset[i]<gamma){
        list[num_local_node]=i;
        mask[i]=1;
        num_local_node++;
    }
}

for(i=0;i<num_element;i++){
    for(j=0;j<num_node_element;j++){
        if(mask[n[j]]==1)node_in_domain++;
    }
    if(node_in_domain==num_node_element){
        list_element[num_local_element]=i;
        num_local_element++;
    }
}
```

Figure 4.6. Pseudo-code to allocate the local domain (Localization of XFEM)

The algorithm for localization of XFEM is explained with the pseudo-code of the figure 4.6. The every node is scanned to distinguish whether the node is in the local domain or outside. When the level set value of the node is smaller than six times the mesh size, the node is indexed to 1 and included in a list for localization. The elements in the local domain are distinguished by checking the number of nodes indexed to 1 among the nodes composing each element. The elements are included in a list of the local domain when the number of nodes indexed to 1 is equal to the number of nodes composing each element.

```
boundary=5*size_mesh;
for(i=0;i<num_node;i++){
    if(levelset[i]>boundary&&levelset[i]<gamma){
        duplication=0;
        for(j=0;j<num_vent;j++){
            if(list_vent[j]==i){
                duplication=1;
                break;
            }
        }
        if(duplication==0){
            air_bc_list[num_air_bc]=i;
            num_air_bc++;
        }
    }
}
for(i=0;i<num_vent;i++){
    if(mask[list_vent[i]]==1){
        air_bc_list[num_air_bc]=list_vent[i];
        num_air_bc++;

    }
}
```

Figure 4.7. Pseudo-code to make a node list on the boundary

The figure 4.7 is the algorithm to make a list of the boundary. The list is used to put the boundary condition to the vent pressure. As scanning the level set value of each node, the nodes placed on the boundary of the local domain is identified. In order to prevent existence of duplicate nodes in the list, the code checks whether the node is in a list of nodes on the vent. And, when the nodes on the vents are in the local domain, the nodes are included in the list of the boundary.

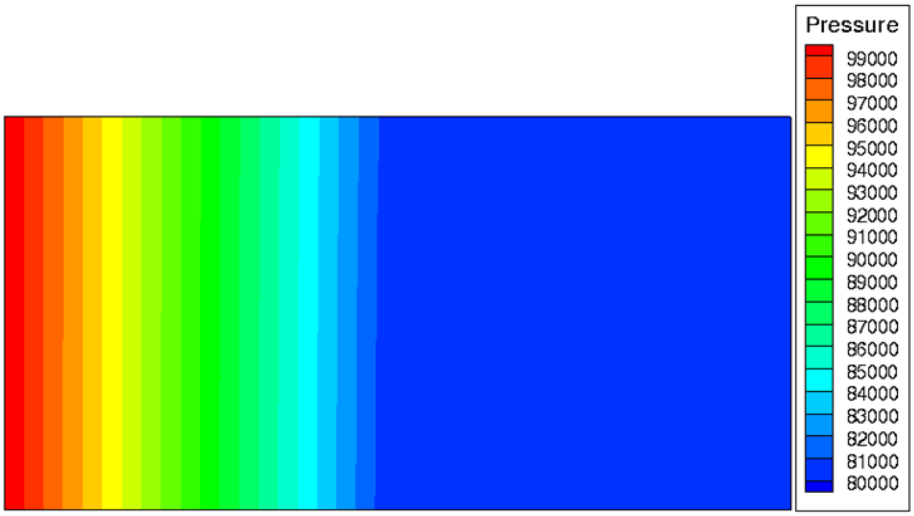


Figure 4.8. Pressure contour
(Localization of XFEM)

In the figure 4.8, the pressure contour is obtained by the calculation with localization of XFEM. The contour is almost same with the computation result (Figure 4.4) without the localization.

4.3. Efficiency Test

The efficiency of the localization algorithm is shown by using simple examples: channel and radial flow models. The models have different mesh size in order to check the effect by the increase of the DOF size. The modeling information and the calculation results are explained in Section 3.4.3, therefore, this section discusses only about the computation time. The machine used in this calculation is Intel (R) Core(TM) i7 CPU 950 (3.07 GHz, Quad-core). The operation system is Windows 7 Professional K for 64 bit machine.

The first example is the channel flow model of the figure 3.14. The computing times of the models are shown in the table 1. The first column indicates the number of node (DOF). The data of the second column is the computing time by the original code which does not apply the localization algorithm. The third column is the time with the localization. The calculation time by applying the localization of level set method is the upper data. And, the below date is the time with localization of XFEM and level set method. The forth column is the percentage of the reduced time by the localization algorithm. The reduction is obtained as :

$$\text{Reduction} = \frac{a-b}{a} \times 100 \qquad (4.3)$$

, where a and b are the computing times without localization and with localization, respectively.

The reduction percentages of the three cases by localization of level set method are about 23%, 27%, and 40%, respectively. Furthermore, their reductions by localization of XFEM are about 36%, 44%, and 55%.

Table 1. Comparison of computing times (A channel flow model)

No. of Node	Computing Time			Reduction
	Without Localization (a)	With Localization (b)		
861	37.5 sec.	Level-Set	29.0 sec.	22.7 %
		XFEM + Level-Set	24.1 sec.	35.7 %
2556	211.4 sec.	Level-Set	154.0 sec.	27.2 %
		XFEM + Level-Set	118.0 sec.	44.2 %
7381	1236.8 sec.	Level-Set	742.5 sec.	40.0 %
		XFEM + Level-Set	561.1 sec.	54.6 %

The second example is the radial flow model as shown in the figure 3.15. Table 2 is the computing time. The numbers of nodes for three cases are 2082, 7295, and 8040 nodes, respectively. The reductions of the computing times with localization of level set method are approximately 25%, 30%, and 37%, respectively. The reduction rates with localization of XFEM and level set method are 37%, 42%, and 40%, respectively.

Table 2. Comparison of computing times (A radial flow model)

No. of Node	Computation Time			Reduction
	Without Localization(a)	With Localization (b)		
2082	185.2 sec.	Level-Set	139.8 sec.	24.5 %
		XFEM + Level-Set	117.0 sec.	36.8 %
7295	1349.5 sec.	Level-Set	943.4 sec.	30.1 %
		XFEM + Level-Set	782.8 sec.	42.0 %
12480	3122.6 sec.	Level-Set	1965.9 sec.	37.0 %
		XFEM + Level-Set	1665.2 sec.	46.7 %

The reduction tendency of the computing time looks similar in the channel flow and the radial flow cases. That is to say, the reduction rate is increased with the bigger number of nodes. The reason might be that the reduction of the computational domain in the model of the bigger number of nodes is more than the reduction in the smaller nodes model. The computational amount of level set calculation is more than the one of pressure calculation. Therefore, the effect by localization of level set method is larger than the one by the localization of XFEM. After all, the localization of XFEM maximizes the reduction rate with the localization of level set method. From the test of the computing time, it could be definitely realized that the computational efficiency is increased by using the localization.

Chapter 5. Applications

Four application examples are treated in this chapter. The first is to analyze the filling stage of a complex edge mold with a hole. The next examples show a race tracking effect and macro void formation. The last is the analysis of the RTM process to manufacture a composite wind turbine blade. In addition, this study is implemented to the total GUI tool for modeling and analyzing of RTM process.

5.1. Plate with Complex Edge

A plate with complex shaped edges and a hole is chosen, and the analysis result is compared with the experimental. Since the flow pattern during mold filling is affected by the structure geometry, the filling pattern of the resin in the complex geometry mold changes a lot.

The experimental result and the modeling information are acquired from the research of Akbar Shojaei et al. [22].

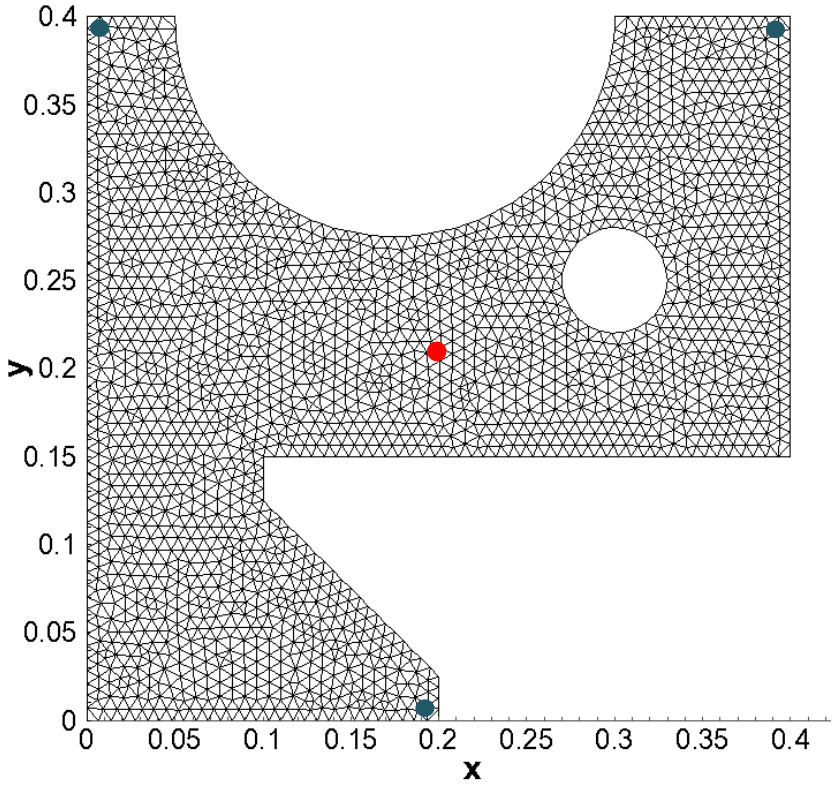
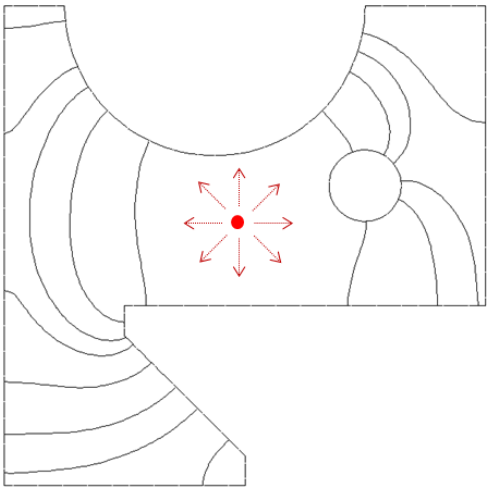


Figure 5.1. Geometry and mesh of the plate

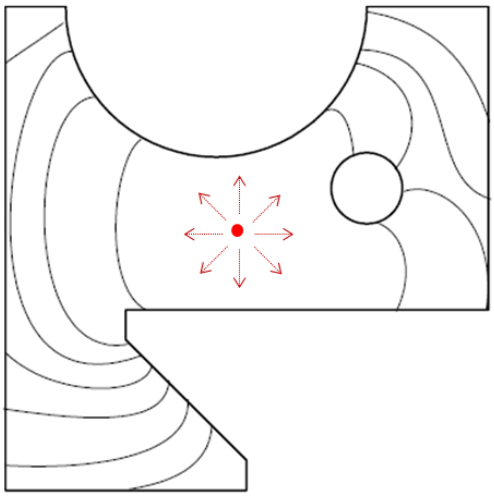
The geometry and mesh is shown in the figure 5.1. The width and height are equal to 0.4 m, and its thickness of the plate is equal to 6.5 mm. The injection gate is located at the center as shown by the red point. The diameter of the injection gate is 4 mm. The vents are placed at the edges as marked by three blue points. All the edges of the mold are sealed by silicone to prevent the edge effect. The meshed plate has 2204 nodes and 4111 triangular elements. SAE 40 motor oil is used as resin. Its viscosity is equal to 0.3 Pa·s at room temperature (25 ℃). The permeability and porosity of the fiber preform are 6.83×10^{-9} m^2 and

0.81, respectively. The injection pressure is set to 35 KPa. And, the vent pressure and initial pressure in the mold are set to zero.

The total filling time obtained in this study is equal to 123 sec. The experimental and numerical total filling times obtained by Akbar Shojaei et al. [22] are equal to 117 sec. and 126 sec., respectively. The difference between the experimental result and the numerical one obtained in this study is equal to 4.3 %. In comparison with the numerical solution proposed by Akbar Shojaei et al. [22], whose difference is equal to 7.7 %, the result of this study is more similar to the experimental one.



(a) Obtained by methods



(b) Given by experiment

Figure 5.2. Comparison of results on 2D complex geometry plate

(Position of flow front at every 15 sec.)

In the figure 5.2 (a) shows the location of the flow front during the filling process obtained in this work, in comparison with the experimental ones given by Akbar Shojaei et al. [22] (Figure 5.2 (b)). The flow front locations are plotted at every 15 seconds.

5.2. Thin and Complex Shape Structure

A thin and complex shaped structure is used to analyze the race-tracking effect and macro void formation in the mold. The numerical simulations of the race-tracking effect and macro void formation in the mold could be useful to correctly design process variables, such as the injection/vent locations and their numbers. If there is no special treatment, the race-tracking effect appears in the actual injection procedure due to small space between the mold and the fiber preform edges. Macro void formation in the mold can occur due to the incorrect design of the variables such as the injection/vent positions and their numbers.

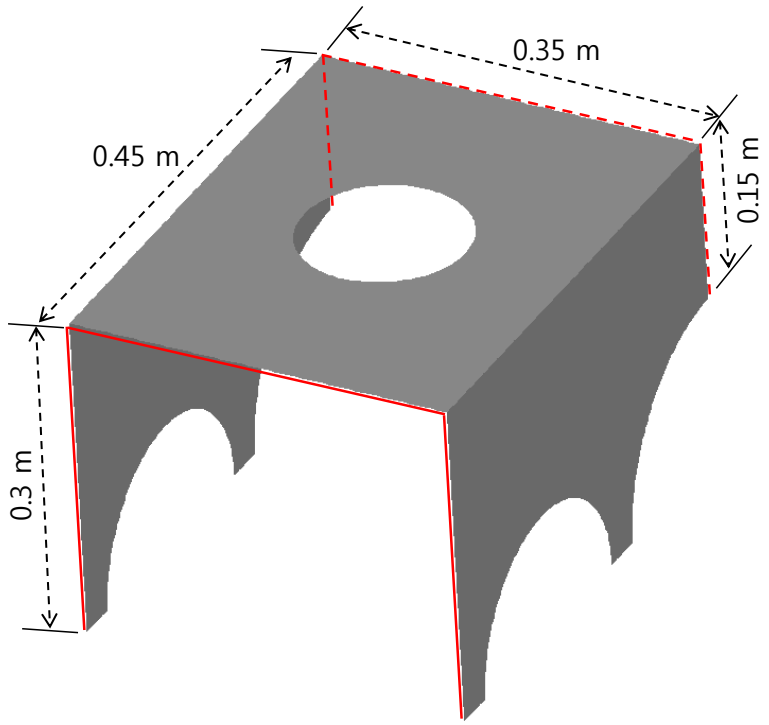


Figure 5.3. Geometry of the thin and complex shape structure

The geometry is shown in the figure 5.3. The diameter of the hole in the example is 0.11 m. The solid line and the dashed line represent the position of the injection gates and the vents, respectively. The material properties and boundary conditions are: resin viscosity = 0.03 Pa·s, preform permeability (isotropy and constancy) = 6.21×10^{-11} m^2, porosity = 0.4, injection pressure = 0.3 MPa, vent pressure = 0. The type of elements used in this analysis is 3-nodes triangle.

5.2.1 Race Tracking Effect

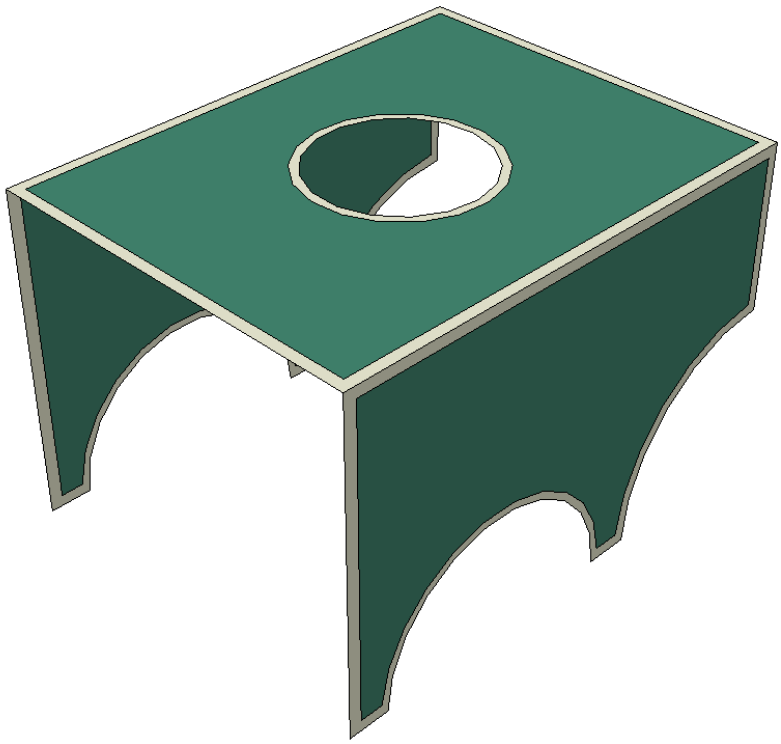


Figure 5.4. Two different permeability regions
for the race-tracking effect

As shown in the figure 5.4, to produce the race-tracking effect, a narrow band at the edge, of which the width is constant and equal to 10 mm, has a high permeability, equal to 6.21×10^{-10} m^2. The meshed model has 3684 nodes and 7027 elements.

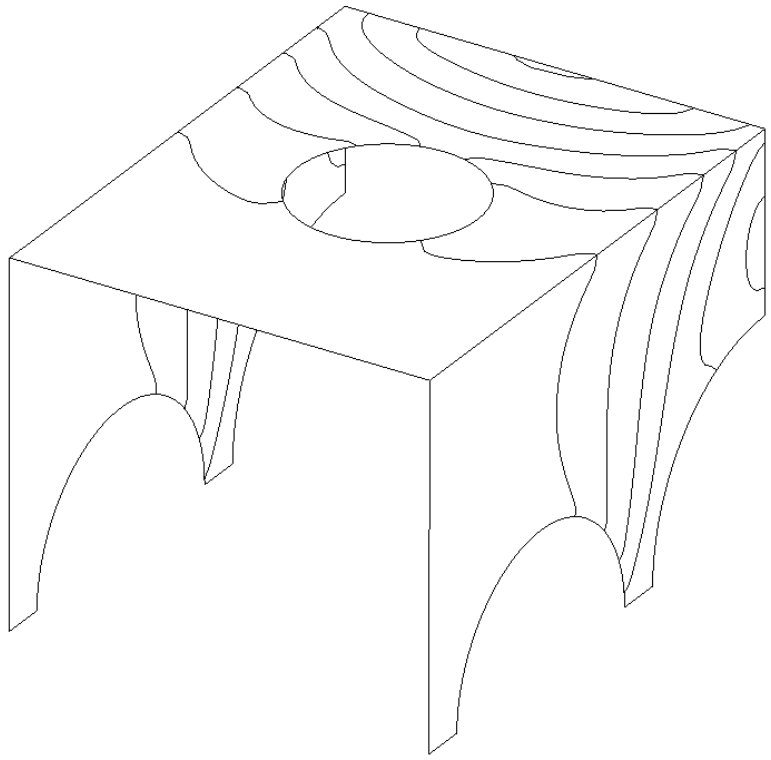


Figure 5.5. Results of the race tracking effect model
(Position of flow front at every 5 sec.)

The figure 5.5 shows the resin flow front obtained in this study. The solid line represents the positions of the resin fronts at every 5 sec. during the mold filling. It shows that the resin flow fronts near the edge have curvature due to the race-tracking effect. In addition, the dividing and merging of the resin fronts are well-simulated.

5.2.2 Macro Void Formation

To show the formation of macro voids during the mold filling, the hole is removed at the upper side. And without the race-tracking effect, the permeability of the narrow band at the edges is as same as the value in the inside of the mold. The other geometry and injection/vent gates are same with the figure 5.3. This model has 3812 nodes and 7328 3-node elements.

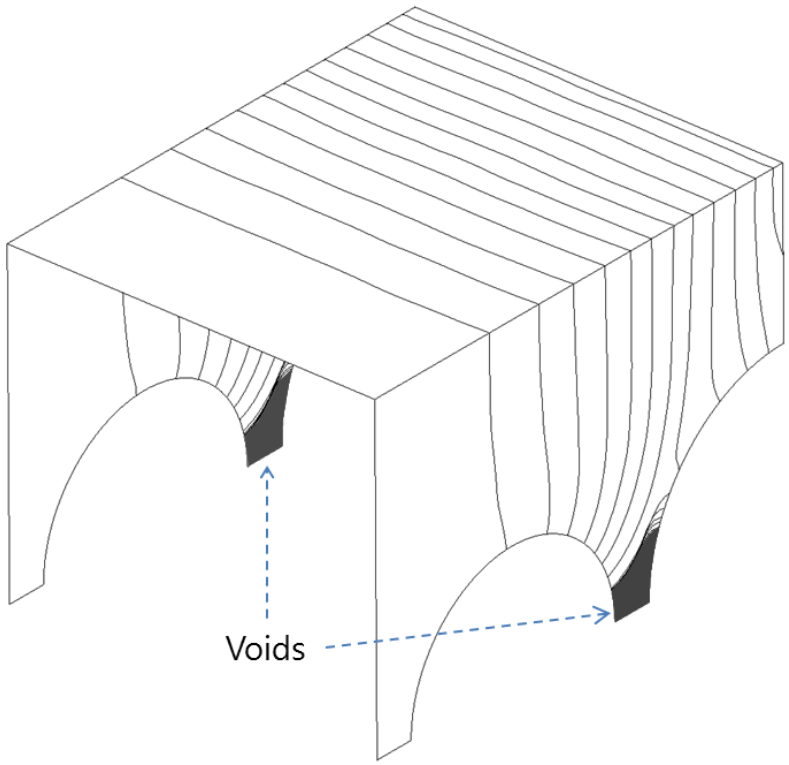


Figure 5.6. Formation of macro voids
(Position of flow front at every 5 sec.)

The computation result is plotted in the figure 5.6. It shows the position of the resin fronts every 5 sec. during filling. The macro voids

were trapped at the corners marked by the dark color in the figure 5.6. The results show that our program could well catch the formation of macro voids.

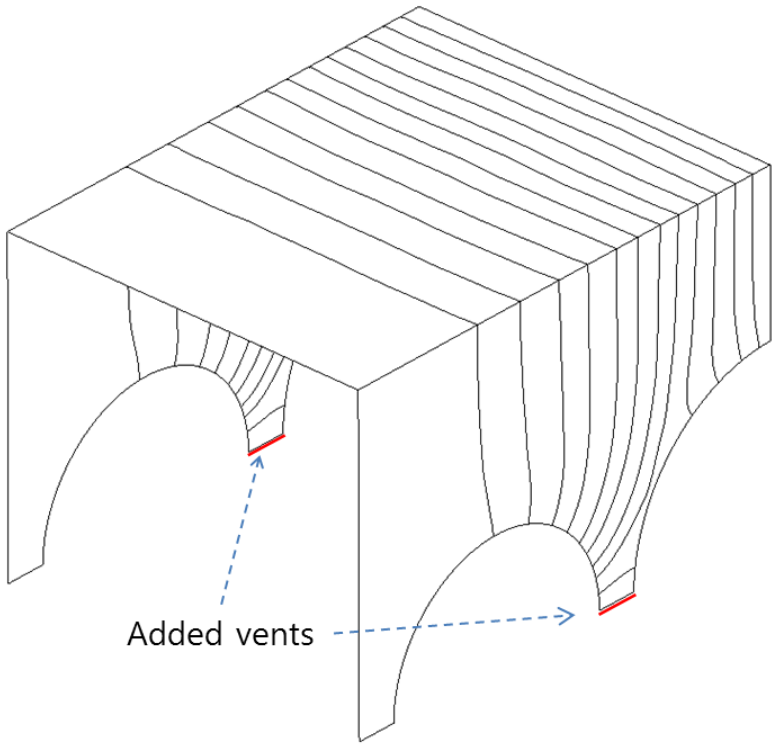


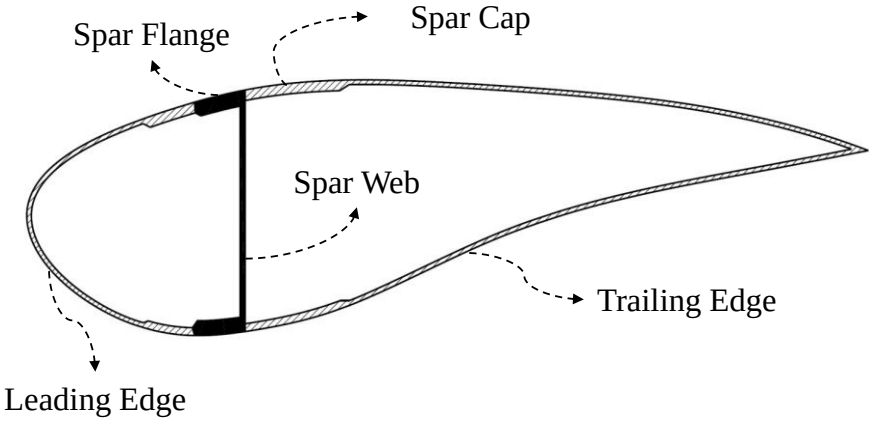
Figure 5.7. Insertion of vents
(Position of flow front at every 5 sec.)

The formation of macro voids can be prevented by installing vents. As shown in the figure 5.7, additional vents, marked by the red lines, is installed at the region of the trapped macro voids. The flow fronts are plotted at every 5 sec. in the figure 5.7. Because of the added vents, the macro voids at the corners are not captured at all. This analysis shows an important role of numerical simulation for the process design.

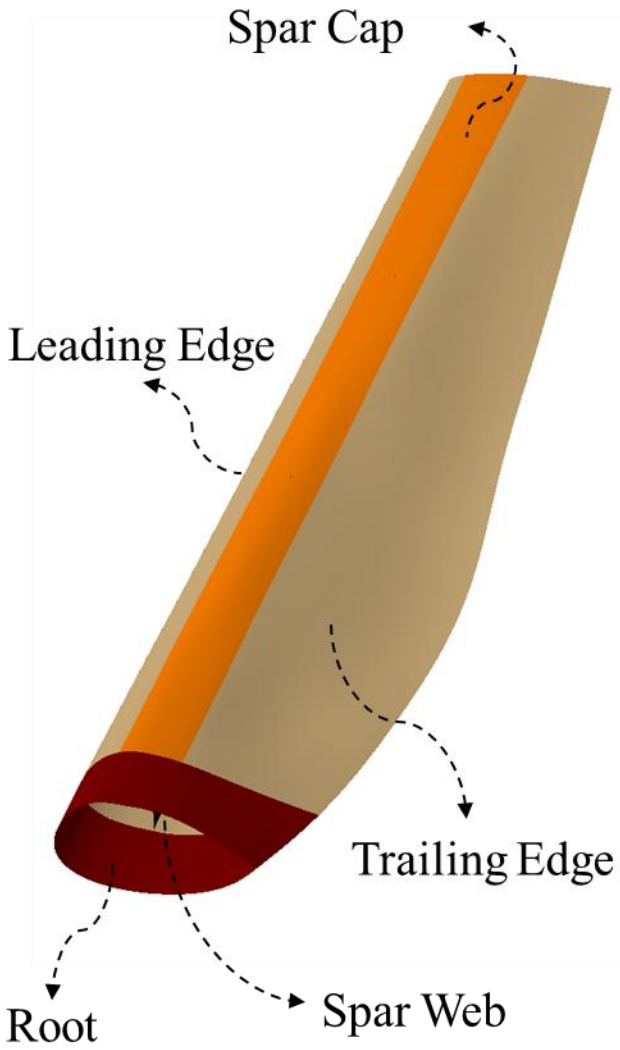
5.3. Wind Turbine Blade

Manufacture of composite wind turbine blades by the RTM process offers the less labor cost and higher fiber volume fraction than the hand lay-up process [67]. However, for manufacturing large scale blades by RTM process, the mold design might be very expensive because it is difficult to predict the resin flow in the closed mold during the filling. Therefore, it is necessary to numerically simulate the mold filling stage in order to minimize the cost of the mold design.

A composite blade for the Atlantic Orient Corporation (AOC) 15/50 wind turbine [67-69] is treated to simulate the resin flow during the RTM process. The modeling information such as the geometry and material properties are obtained from the report of Cairns and Rossell [67]. The length of the composite blade is about 6 m. The coordinates to draw the blade model is explained in Appendix B.



(a) Cross section



(b) Whole structure

Figure 5.8. AOC 15/50 wind turbine blade

The cross-section and whole structure of the composite blade is shown in the figure 5.8. The blade is composed by three sections as web, spar cap, flange, leading edge and trailing edge. The each section has different fiber permeability because the lay-up is different. The details about the lay-up are explained in Cairns and Rossell [67].

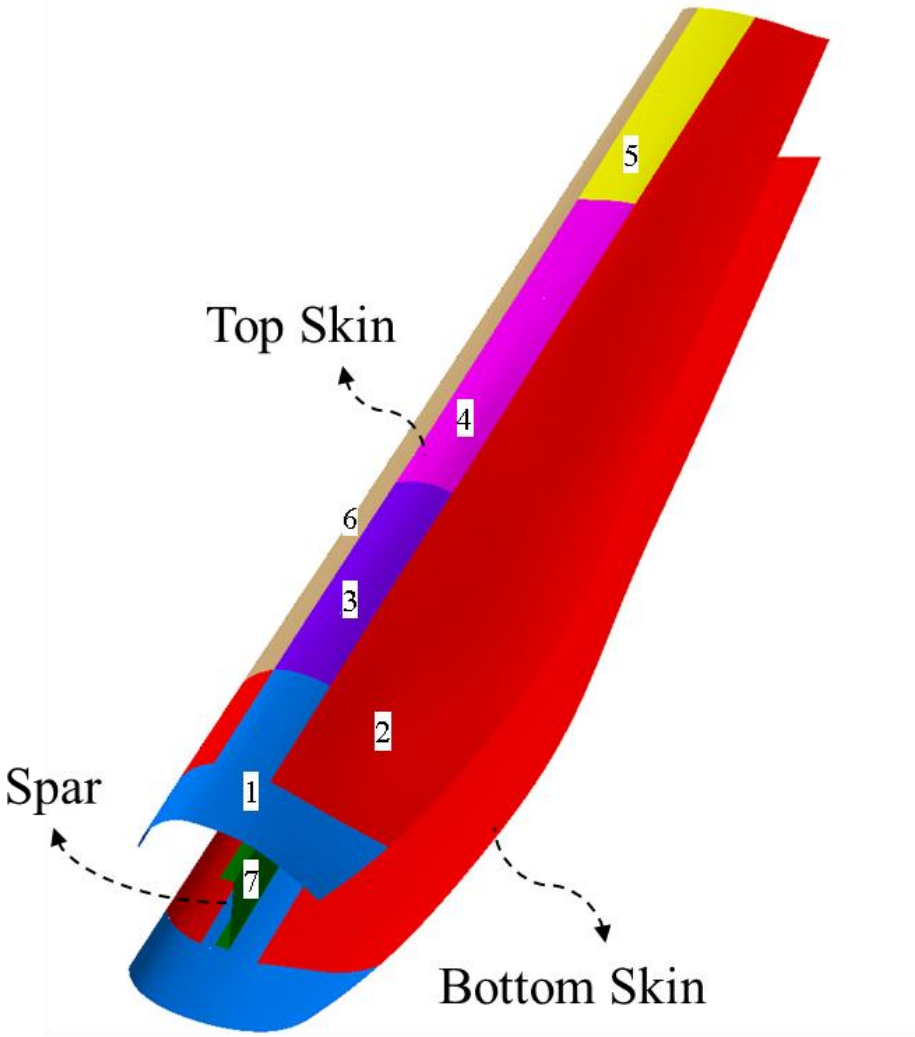


Figure 5.9. Three parts and seven sections of the composite wind turbine blade model

The blade is made by assembling three parts: top skin, spar, and bottom skin, as shown in the figure 5.9. Each part is individually manufactured by RTM process.

Table 3. Permeability of seven sections in the composite wind turbine blade model

Section No.	Section Name	Permeability (m^2)	
		K_X	K_Y
1	Root + Spar Cap	2.32×10^{-10}	7.05×10^{-11}
2	Trailing Edge + Leading Edge	1.78×10^{-10}	7.06×10^{-11}
3	Spar Cap	1.97×10^{-10}	6.64×10^{-11}
4	Spar Cap	1.52×10^{-10}	5.90×10^{-11}
5	Spar Cap	9.31×10^{-11}	4.46×10^{-11}
6	Leading Edge	1.28×10^{-10}	6.13×10^{-11}
7	Spar	2.02×10^{-10}	7.79×10^{-11}

There are seven sections masked by different colors. The sections have different permeability values. Their permeability are anisotropic and remarked in the table 3. The resin viscosity for the three models is 0.195 Pa·s. And, their injection pressure is 800kPa and constant during the mold filling. The type of elements is 3-nodes triangle.

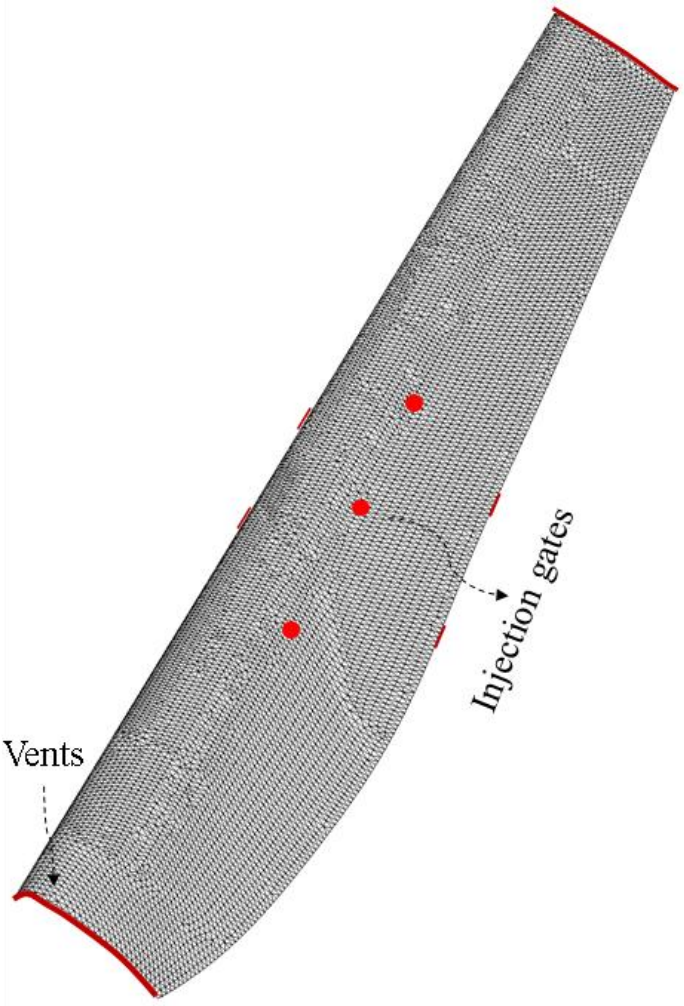


Figure 5.10. Mesh and positions of injection gates & vents on the top skin

At first, the top skin is simulated. The top skin is meshed as the figure 5.10. The mesh is composed by 6848 nodes and 13054 elements. As shown in the figure, the injection gates are located at the three red points at the middle. The vents, marked by the red line, are place at the two end of the structure. And, additional vents are installed at the edge, marked by red lines and merging two flow fronts, in order to prevent the capture of the air voids.

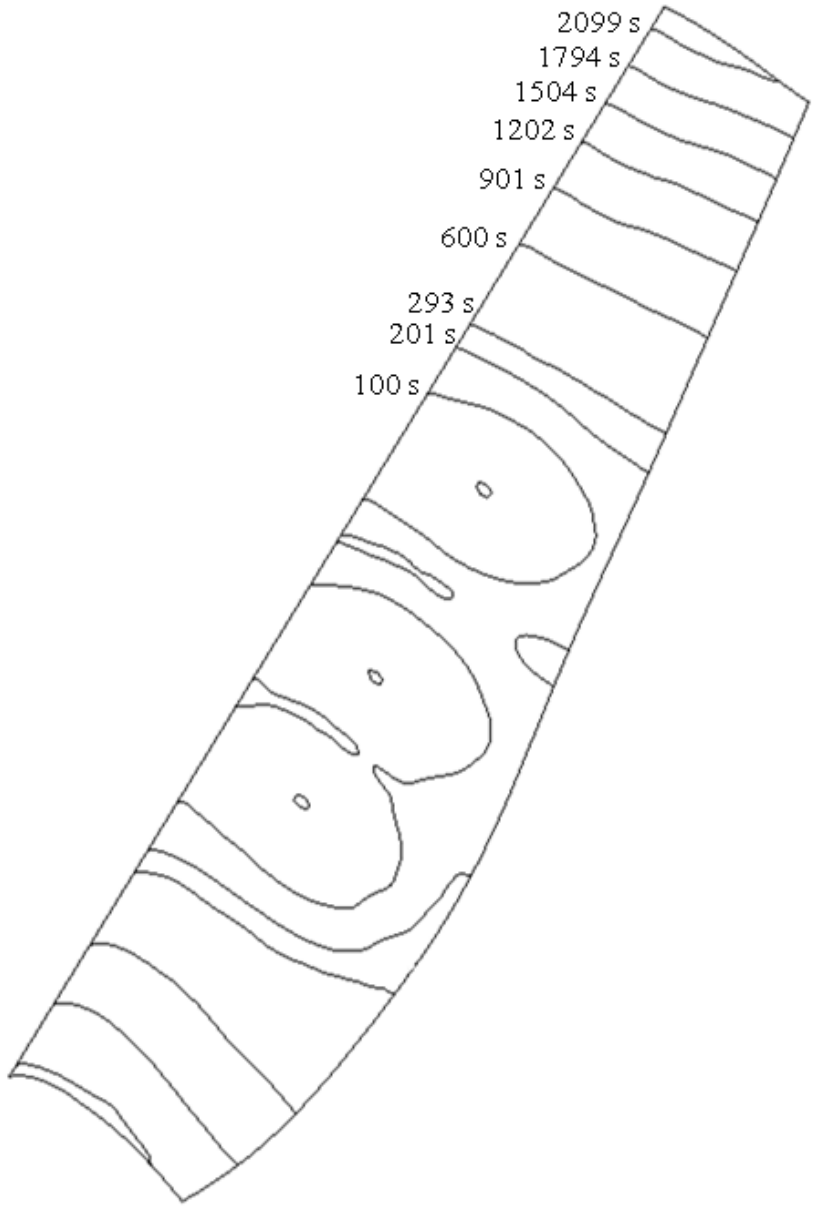


Figure 5.11. Resin flow pattern during the mold filling of the top skin

The filling pattern calculated by numerical simulation is shown in the figure 5.11. The total filling time is about 38 minutes. The resin flow pattern during the mold filling is shown in the figure. The lines on the blade mean the position of the flow fronts at an interval of about

five minutes during the mold filling. The fronts expanded from neighbor injection gates are merged at about 100 sec. and 200 sec., respectively. The fronts at the trailing edge are reached earlier at the both ends because the permeability (especially, K_y) of trailing edge is larger than leading edge and spar cap. Therefore, the pattern of resin flow fronts seems to be reasonable.

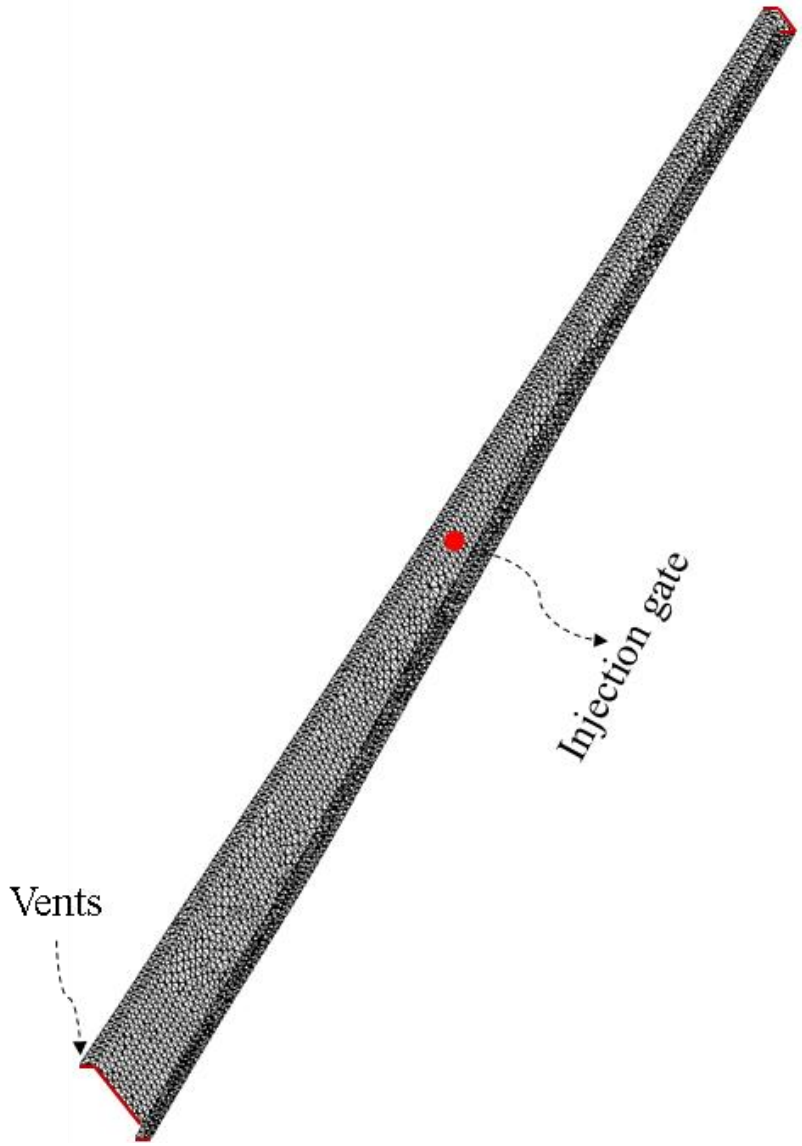


Figure 5.12. Mesh and positions of injection gates & vents on the spar

The second part is a spar of the blade as shown in the figure 5.12. The mesh is composed by 2447 nodes and 4442 elements. The injection gate is located at the middle point of the structure. The vents, marked by the red line, are place at the two end of the spar.

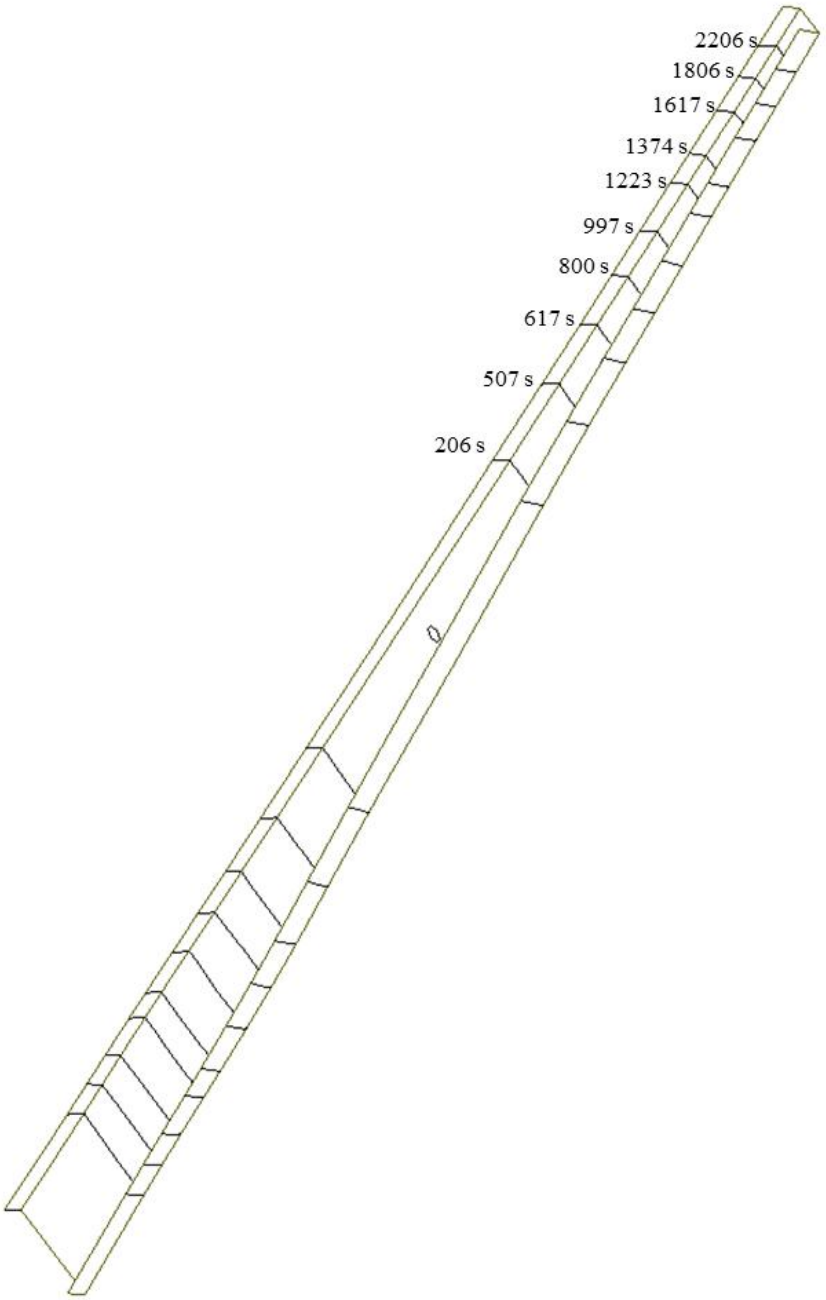


Figure 5.13. Resin flow pattern during the mold filling of the spar

For showing the simulation result of the spar model, the resin flow front at an interval of about 100 seconds is plotted in the figure 5.13. The flow fronts are regularly expanded from the middle injection gate because the spar has uniform permeability and the resin in injected from one spot with constant pressure. Therefore, the filling pattern of the spar model is reasonably described in this study. The total filling time of the spar model is about 45 minutes.

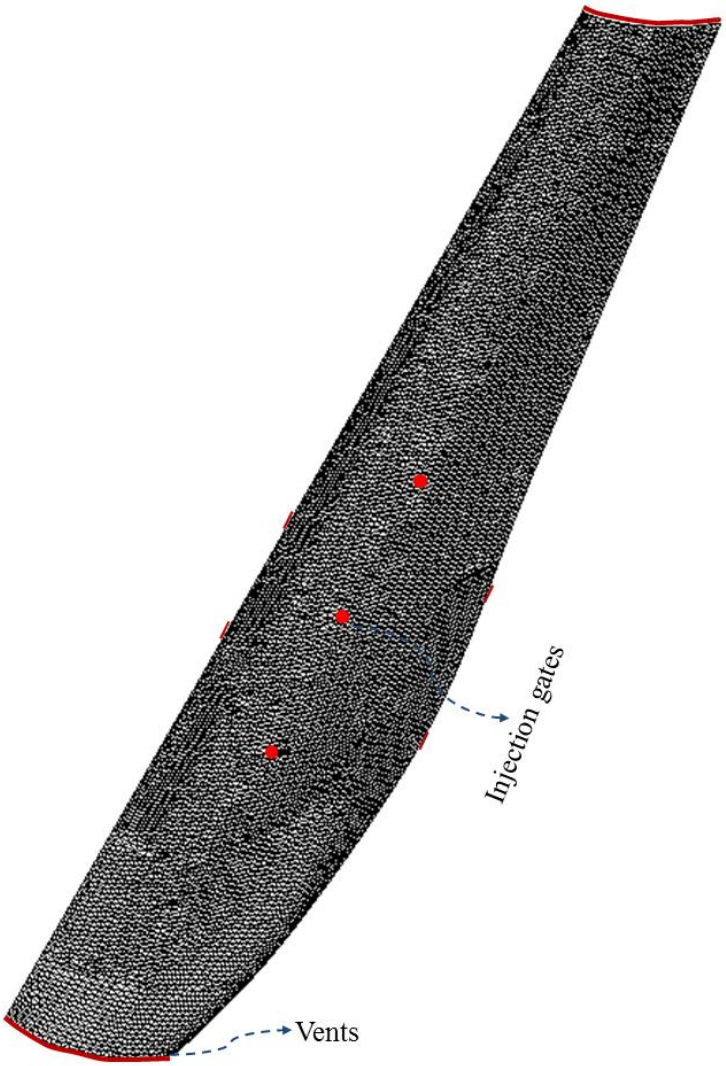


Figure 5.14. Mesh and positions of injection gates & vents on the bottom skin

Finally, the bottom skin of the blade is treated. The figure 5.14 shows the mesh condition of the model and its boundary information such as injection gates and vents. The total number of nodes and elements are 7043 and 13436, respectively. The position of injection gates and vents are similar with them of the top skin model. As shown and explained in the figure 5.9 and table 1, the model is composed by six kinds of permeability.

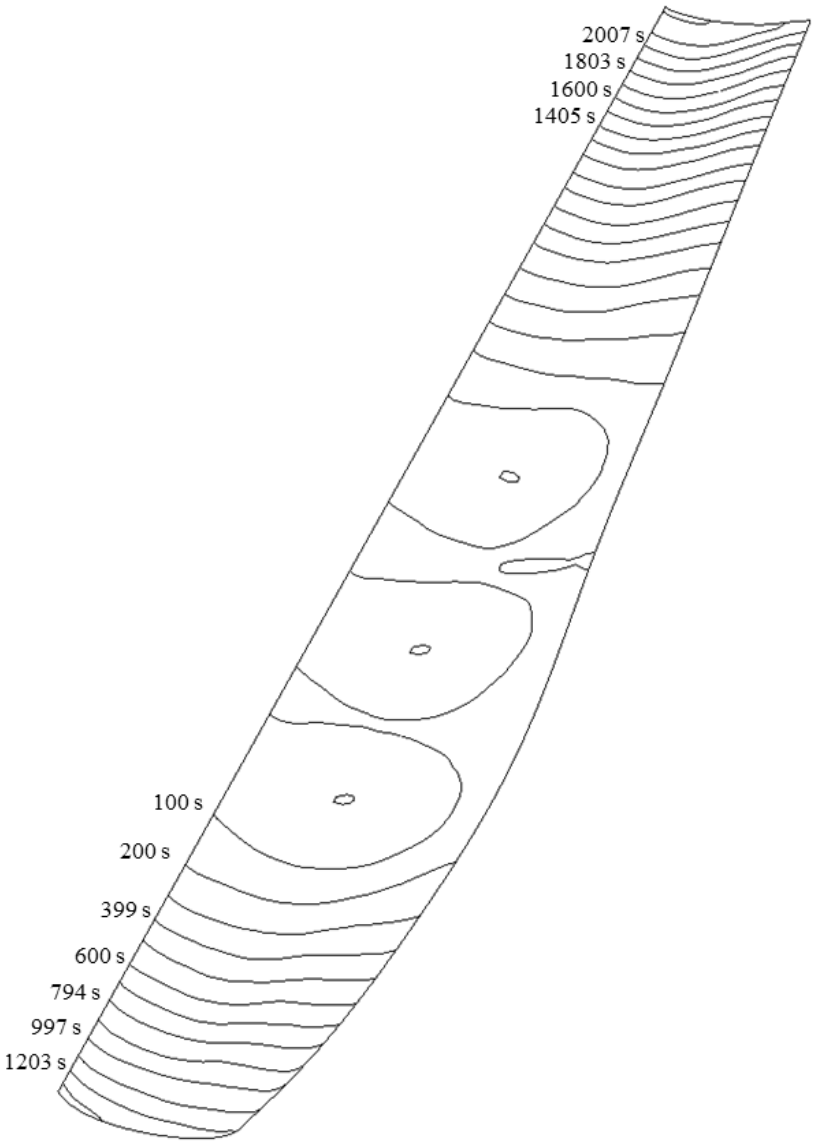


Figure 5.15. Flow pattern during the mold filling of the bottom skin

The simulation result of the bottom skin model is shown in the figure 5.15. The total filling time is approximately 37 minutes. The flow front positions are plotted at the filling time written in the figure. The resin flow front pattern is assembled with the one of the top skin model because both of them have similar permeability and boundary condition. The resin flow of the trailing edge section is faster than the one of the leading edge part. And, the merged motion of flow fronts is well simulated. The slight difference of the flow front patterns and filling time between both models comes from their a little different shape. In general, the resin front pattern of this example is reasonably represented by the numerical simulation.

5.4. Diamond/RTM

Our algorithm, proposed for numerical simulation of the RTM process, is realized by embedding an option for modeling and analysis of RTM process in Diamond. Diamond is developed by Aerospace Structures Laboratory in Seoul National University. It is an integrated finite element program for modeling and analyzing general structural problems in serial and parallel computing environment.

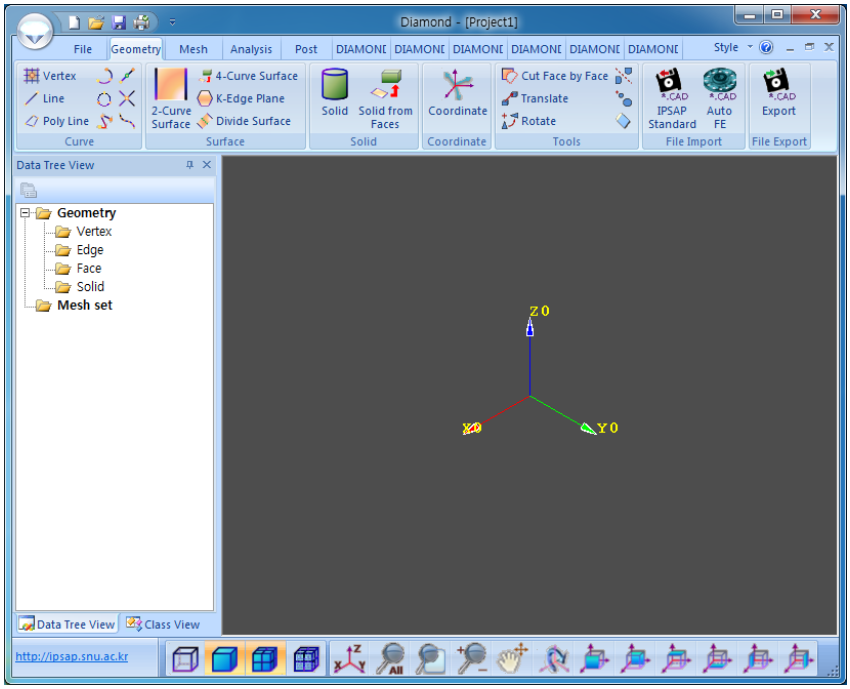


Figure 5.16. Features of Diamond

The figure 5.16 shows the features of Diamond. The convenient functions of Diamond are used to generate geometry and mesh of models for RTM process simulations. In Diamond, the special functions for modeling the RTM process are embedded.

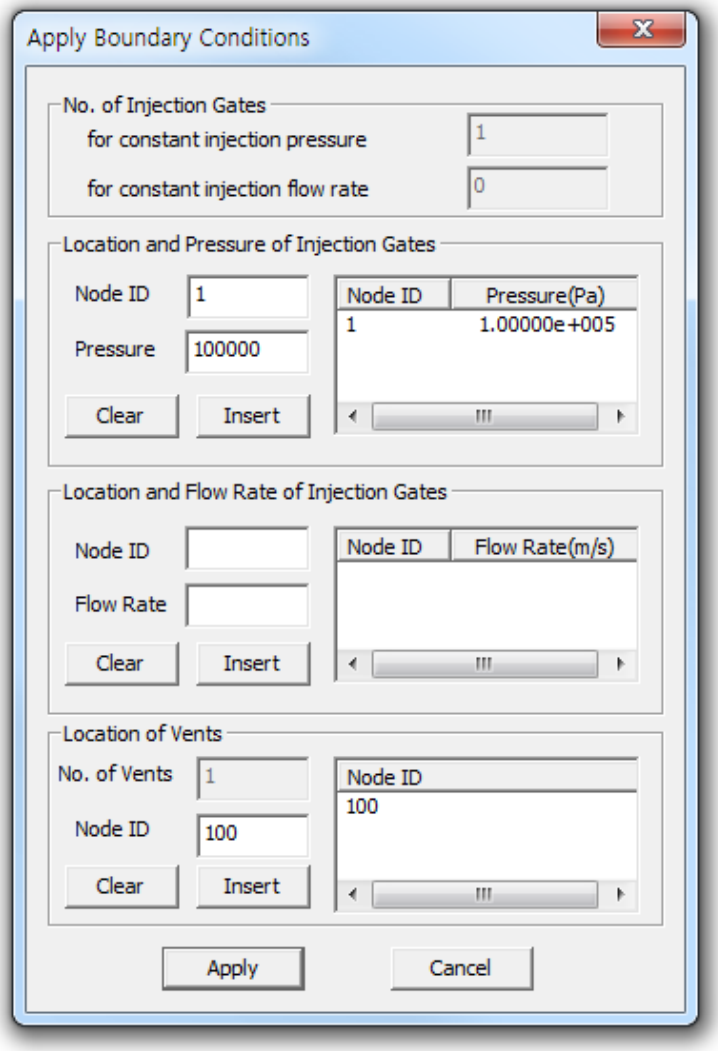


Figure 5.17. Setting boundary conditions
(Injection gates & vents)

The boundary conditions are possible to set by filling the entry in the diagram as the figure 5.17. The entry for the boundary conditions is the information about injection gates and vents. The locations of injection gates and vents are entered by clicking on the corresponding nodes with the mouse.

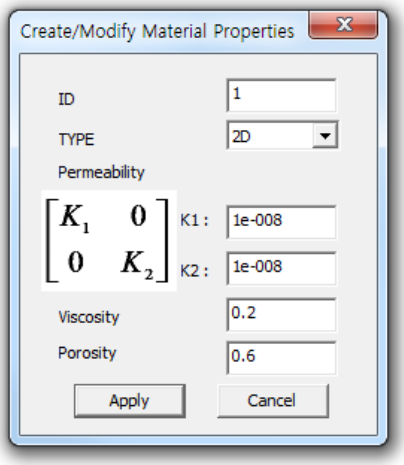


Figure 5.18. Input of material properties

The figure 5.18 is the diagram for input of material properties such as permeability, viscosity, and porosity. If the model is composed by multiple permeability values, it is possible to input them as changing the ID number.

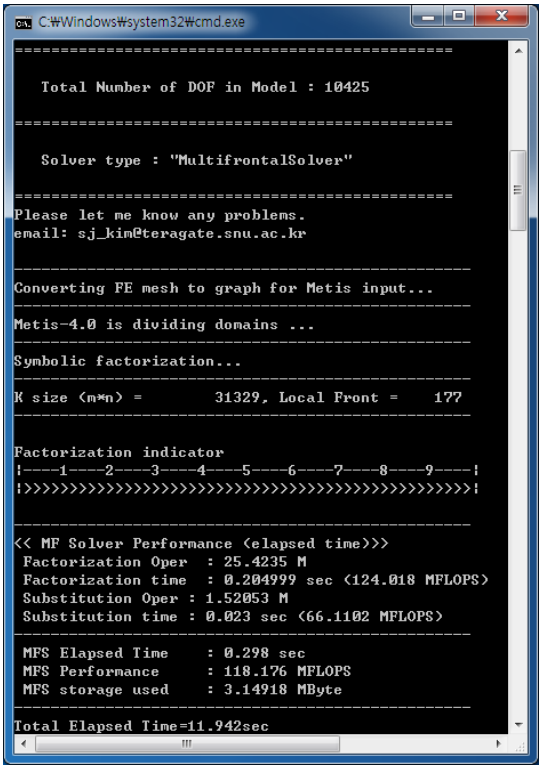


Figure 5.19. Computation information of model

After finishing the modeling, the computation is launched. The window in the figure 5.19 appears to give the computation information of the model during the calculation.

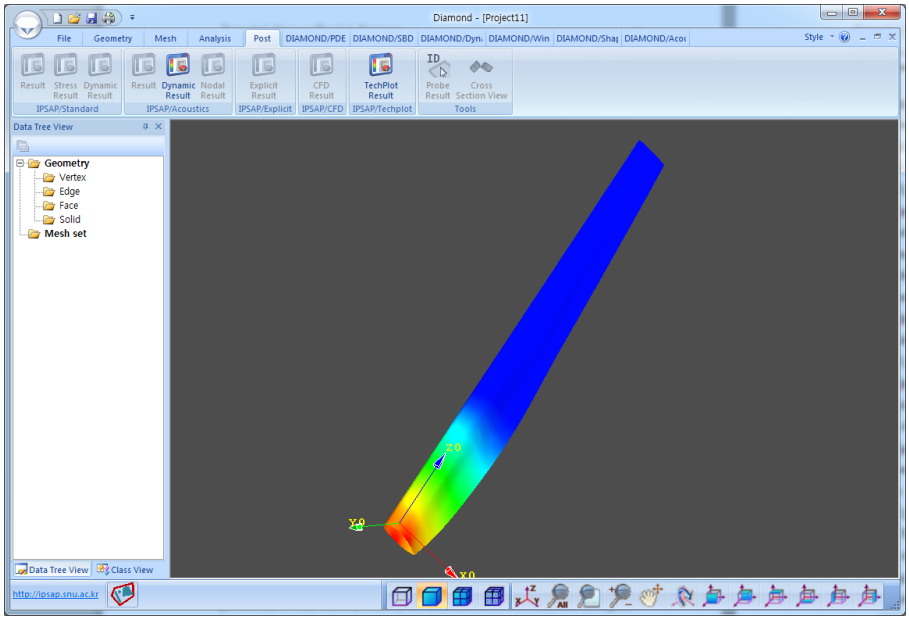


Figure 5.20. Drawing a contour plot of the analysis result in Diamond

The analysis result is able to check by using the post-processing part of Diamond. The figure 5.20 shows to draw the pressure contour of a omposite blade model.

Chapter 6. Conclusions

In this study, the analysis of RTM process is established using XFEM combined with the level set method, so as to track more accurately the resin flow front during the mold filling. The most important advantage of this approach is that it is possible to describe the discontinuity of the pressure gradient at the resin flow fronts. The enriched shape functions are defined by the level set function and reflect the pressure distribution according to the flow front's position. Furthermore, the number of DOFs is slightly increased because the enrichment is just applied near the front. The position of the resin front is determined by solving the transport equation written with the level set function. The equations for transport of flow front and reinitialization of the level set value are solved by an implicit characteristic Galerkin FEM. This study is validated through some examples in comparison with analytical results found in the literature. The differences between the obtained results and the analytical solutions are little.

Additionally, the localization of XFEM and level set method is applied to increase computing efficiency. The localization of XFEM is newly proposed in this study for applying in RTM process simulation. By using the localization, the computing time is dramatically reduced.

The separating and merging of the resin flow front are well simulated on the analysis of a plate with complex shaped edges. The analysis on the plate of thin and complex geometry shows that this study could describe race tracking effects and macro void formation. The composite wind turbine blade is treated to simulate the mold filling stage during the RTM process. The filling patterns during the mold filling of the composite blade are well-described. Finally, this study is realized as GUI tool for RTM process simulation (Diamond/RTM). The every procedures for modeling and analyzing the RTM process might be conveniently done by the Diamond/RTM.

From this study, we identified the XFEM is useful to analyze the resin flow motion in the mold during RTM process. The methods proposed in this work make the analysis accurate and efficient. In the future, this study can be adopted to investigate efficient optimization of the design variables in the mold filling stage because the proposed methods give high efficiency and accuracy. This study is proper for analysis of the mold filling stage in RTM process and meaningful for the problem finished the filling before the curing reaction. This study can be developed to total simulation program for RTM process if the analyses of heat transfer and curing reaction are included in the future.

Reference

1. R. Lin, L.J. Lee, and M. Liou, Non-isothermal mold filling and curing simulation in thin cavities with preplaced fiber mats, *International Polymer Processing*, Vol. 4. No.4, P. 356-369 (1991)

2. K.N. Kendall, C.D. Rudd, M.J. Owen, and V. Middleton, Characterization of the resin transfer moulding process, *Composites Manufacturing*, Vol. 3, No. 4, P. 235-249 (1992)

3. Francois Trochu and Raymond Gauvin, Limitations of a boundary-fitted finite difference method for the simulation of the resin transfer molding process, *Journal of Reinforced Plastics and Composites*, Vol. 11, No. 7, P. 772-786 (1992)

4. Gerd Morren, Sven Bossuyt, and Hugo Sol, 2D permeability tensor identification of fibrous reinforcements for RTM using an inverse method, Composites Part A: Applied Science and Manufacturing, Vol.39, No. 9 (2008)

5. Wen-Bin Young, Development of a helicopter landing gear prototype using resin infusion molding, Journal of Reinforced Plastics and Composites, Vol. 28, No. 7, P. 833-839 (2009)

6. S. Subbiah, D.L. Trafford, and S.I. Guceri, Non-isothermal flow of polymers into two-dimensional, thin cavity molds: a numerical grid generation approach, *International Journal of Heat and Mass Transfer*, Vol. 32, No. 3, P. 415-434 (1989)

7. Hua Tan and Krishna M. Pillai, Numerical simulation of reactive flow in liquid composite molding using flux-corrected transport (FCT) based finite element/control volume (FE/CV) method, *International Journal of Heat and Mass Transfer*, Vol. 53, No. 9-10, P. 2256-2271 (2010)

8. Murat Danisman, Goker Tuncol, Alper Kaynar, and E. Murat Sozer, Monitoring of resin flow in the resin transfer molding (RTM) process using point-voltage sensors, Composite Science and Technology, Vol. 67, No. 3-4, P. 367-379 (2007)

9. Kuang-Ting Hsiao and Suresh G. Advani, Flow sensing and control strategies to address race-tracking disturbances in resin transfer molding. Part I: design and algorithm development, Vol. 35, No. 10, P. 1149-1159 (2004)

10. David Rouison, Mohini Sain, and M. Couturier, Resin transfer molding of hemp fiber composites: optimization of the process and mechanical properties of the materials, Composite Science and Technology, Vol. 66, No. 7-8, P. 895-906 (2006)

11. Shunliang Jiang, Chuck Zhang, and Ben Wang, Optimum arrangement of gate and vent locations for RTM process design using a mesh distance-based approach, *Composite Part A: Applied Science and Manufacturing*, Vol. 33, No. 4, P. 471-481 (2002)

12. Roopesh Mathur, Bruce K. Fink, and Suresh G. Advani, Use of genetic algorithms to optimize gate and vent locations for the resin transfer molding process, *Polymer Composites*, Vol. 20, No. 2, P. 167-178 (1999)

13. W.B. Young, Gate location optimization in liquid composite molding using genetic algorithms, *Journal of Composite Materials*, Vol. 20, No. 2, P. 1098-1113 (1999)

14. L. James Lee, W. B. Young, and R. J. Lin, Mold filling and cure modeling of RTM and SRIM processes, *Composite Structures*, Vol. 27 No. 1-2, P. 109-120 (1994)

15. Baichen Liu, Simon Bickerton, and Suresh G. Advani, Modeling and simulation of resin transfer moulding (RTM)—gate control, venting and dry spot prediction, *Composite Part A: Applied Science and Manufacturing*, Vol. 27, No. 2, P. 89-163 (1996)

16. Alexis Boccard, Woo Il Lee, George S. Springer, Model for determining the vent locations and the fill time of resin transfer molds, *Journal of Composite Materials*, Vol. 29, No. 3, P. 306-333 (1995)

17. Bruschke M. V. and Advani S. G., A finite-element control volume approach to mold filling in anisotropic porous-media, *Polymer Composites*, Vol. 11, No. 6, P. 398-405 (1990)

18. Moon Koo Kang, Jae Joon Jung, and Woo Il Lee, Analysis of resin transfer moulding process with controlled multiple gates resin injection. *Composites Part A: applied science and manufacturing*, Vol. 31, No. 5, P. 407-422 (2000)

19. Moon Koo Kang and Woo Il Lee, A flow-front refinement technique for the numerical simulation of the resin-transfer molding process. *Composites Science and Technology*, Vol. 59, No. 11, P.1663-1674 (1999)

20. Xiao-Lin Liu, Isothermal flow simulation of liquid composite molding. *Composites Part A: applied science and manufacturing*, Vol. 31, No. 12, P. 1295-1302 (2000).

21. Akbar Shojaei, S. Reza Ghaffarian, and S.M.H. Karimian, Numerical simulation of three-dimensional mold filling process in resin transfer molding using quasi-steady state and partial saturation formulations, *Composites Science and Technology*, Vol. 62, No. 6, P. 861-879 (2002)

22. Akbar Shojaei and S. Reza Ghaffarian, Modeling and simulation approaches in the resin transfer molding process: A review, *Polymer Composites*, Vol. 24, No. 4, P. 525-544 (2003)

23. Akbar Shojaei, S. Reza Ghaffarian and S. Mohammad-Hossien Karimian, Numerical analysis of controlled injection strategies in resin transfer molding, *Journal of Composite Materials*, Vol. 37, No. 11, P. 1011-1035 (2003)

24. F. Trochu, R. Gauvin, and D.M. Gao, Numerical-Analysis of the Resin Transfer Molding Process by the Finite-Element Method. *Advances in Polymer Technology*, Vol. 12, No. 4, P. 329-342 (1993)

25. Francois Trochu, Edu Ruiz, Vincent Achim, and Sofiane Soukane, Advanced numerical simulation of liquid composite molding for process analysis and optimization, *Composites Part A: applied science and manufacturing*, Vol. 37, No. 6, P. 890-902 (2006)

26. W. B. Young, K. Rupel, K. Han, L. James Lee, and Ming J. Liou, Analysis of resin injection molding in molds with preplaced fiber mats. Ⅱ: Numerical Simulation and Experiments of Mold Filling, *Polymer Composites*, Vol. 12, No.1, P. 30-38 (1991)

27. C. W. Hirt and B. D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *Journal of Computational Physics*, Vol. 39, No. 1, P. 201-225 (1981)

28. Min Soo Kim and Woo Il Lee, A new VOF-based numerical scheme for the simulation of fluid flow with free surface. Part I: New free surface-tracking algorithm and its verification, *International Journal for Numerical Methods in Fluids*, Vol. 42, No. 7, P. 765-790 (2003)

29. R. Gantois, A. Cantarel, G. Dusserre, J.-N. Felices, and F. Schmidt, Numerical simulation of resin transfer molding using BEM and level set

method, *International Journal of Material Forming*, Vol. 3, Suppl 1, P. 635-638 (2010)

30. Ching-Long Lin, Haegyun Lee, Taehun Lee, and Larry J. Weber, A level set characteristic Galerkin finite element method for free surface flows, *International Journal for Numerical Methods in Fluids*, Vol. 49, No. 5, P. 521-547 (2005)

31. Danping Peng, Barry Merriman, Stanley Osher, Hongkai Zhao, and Myungjoo Kang, A PDE-based fast local level set method, *Journal of Computational Physics*, Vol. 155, No. 2, P.410-438 (1999)

32. S. Soukane and F. Trochu, Application of the level set method to the simulation of resin transfer molding, *Composites Science and Technology*, Vol. 66, No. 7-8, P. 1067-1080 (2006)

33. J. A. Sethian, Level set methods and fast marching methods 2nd edition, *Cambridge University Press* (1999)

34. G. Pacquaut, J. Bruchon, N. Moulin1, and S. Drapier, Combining a level-set method and a mixed stabilized P1/P1 formulation for coupling Stokes–Darcy flows, *International Journal for Numerical Methods in Fluids*, Vol. 69, No. 2, P. 459-480 (2012)

35. Hong-kai Zhao, T. Chan, B. Merriman, and S. Osher, A variational level set approach to multiphase motion, *Journal of Computational Physics*, Vol. 127, P. 179-195 (1996)

36. Ravikanth Malladi, James A. Sethian, and Baba C. Vemuri, Shape modeling with front propagation: a level set approach, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 2, P. 158-175 (1995)

37. L. L. Zheng and H. Zhang, An adaptive level set method for moving boundary problems: application to droplet spreading and solidification, *Numerical Heat Transfer, Part B*, Vol. 37, No. 4, P. 437-454 (2000)

38. Besson, S.J., Barlaud, M. and Aubert, G., Detection and tracking of moving objects using a new level set based method, Proceedings 15th International Conference on Pattern Recognition, Vol. 3, P. 1100-1105 (2000)

39. Chia-Jung Chang, Jun-Wei Hsieh, Yung-Sheng Chen, and Wen-Fong Hu, Tracking multiple moving boundary objects using a level set method, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 18, No. 2, P. 101-125 (2004)

40. Anton Smolianski, Finite element level set operator splitting (FELSOS) approach for computing two-fluid unsteady flows with free moving interfaces, International Journal for Numerical Methods in Fluids, Vol 48, No. 3, P. 231-269 (2005)

41. Peter D.M. and Spelt, A level-set approach for simulations of flows with multiple moving contact lines with hysteresis, Journal of Computational Physics, Vol. 207, No. 2, P. 389-404 (2005)

42. Chensong Dong, A modified rule of mixture for the vacuum-assisted resin transfer moulding process simulation, *Composite Science and Technology*, Vol. 68, No. 9 P. 2125-2133 (2008)

43. Shunliang Jiang, Li Yang, Shazly Alsoliby, Guofa Zhou, PCG solver and its computational complexity for implicit control-volume finite-element method of RTM mold filling simulation, *Composites Science and Technology*, Vol. 67, No. 16-17, P. 3316-3322 (2007)

44. Feerick R. and Phelan JR., Simulation of injection process in resin transfer molding, *Polymer Composites*, Vol. 18, No. 4, P. 460-476 (1997)

45. B. R. Baliga and S. V. Patankar, A new finite element formulation for convetcion-diffusion problems, Numerical Heat Transfer, Vol. 3, No. 4, P. 393-409 (1980)

46. K. C. Estacio and N. Mangiavacchi, Simplified model for mould filling simulations using CVFEM and unstructured meshes, Numerical Methods in Biomedical Engineering, Vol. 23, No. 5, P. 345-361 (2007)

47. Ted Belytschko, Robert Gracie, and Giulio Ventura, A review of extended/generalized finite element methods for material modeling, *Modeling and Simulation in Materials Science and Engineering*, Vol. 17, No. 4, P. 1-24 (2009)

48. Soheil Mohammadi, Extended finite element method for fracture analysis of structures, Blackwell Publish Ltd. (2008)

49. N. Sukumar, N. Moes, B. Moran, and T. Belytschko, Extended finite element method for three-dimensional crack modeling, *International Journal for Numerical Methods in Engineering*, Vol. 48, No. 11, P. 1549-1570 (2000)

50. N. Sukumar, D.L. Chopp, N. Moos, and T. Belytschko, Modeling holes and inclusions by level sets in the extended finite-element method, *Computer*

Methods in Applied Mechanics and Engineering, Vol. 190, No. 46-47, P. 6183-6200 (2001)

51. Nicolas Moes and Ted Belytschko, Extended finite element method for cohesive crack growth, *Engineering Fracture Mechanics*, Vol. 69, No. 7, P. 813-833 (2002)

52. R. Huang, N. Sukumar, and J.-H. Prévost, Modeling quasi-static crack growth with the extended finite element method Part II: Numerical applications, *International Journal of Solids and Structures*, Vol. 40, No. 26, P. 7539-7552 (2003)

53. H. Ji, D. Chopp, and J. E. Dolbow, A hybrid extended finite element/level set method for modeling phase transformations, *International Journal for Numerical Methods in Engineering*, Vol. 54, No. 8, P. 1209-1233 (2002)

54. A.R. Khoei, A. Shamloo, and A.R. Azami, Extended finite element method in plasticity forming of powder compaction with contact friction, *International Journal of Solids and Structures*, Vol. 43, No. 18-19, P. 5421-5448 (2006)

55. N. Moës, M. Cloirec, P. Cartraud, and J.-F. Remacle, A computational approach to handle complex microstructure geometries, and Engineering, Vol. 192, No. 28-30, P. 3163-3177 (2003)

56. Jack Chessa, Patrick Smolinski, and Ted Belyschko, The extended finite element method (XFEM) for solidification problems, *International journal for numerical methods in engineering*, Vol. 53, No. 8, P. 1959-1977 (2002)

57. Jack Chessa and Ted Belytschko, An extended finite element method for two-phase fluids, *Journal of Applied Mechanics*, Vol. 70, No. 1, P. 10-17 (2003)

58. Jack Chessa and Ted Belytschko, An extended finite element method and level sets for axisymmetric two-phase flow with surface tension, *International Journal for Numerical Methods in Engineering*, Vol. 58, No. 13, P. 2041-2064 (2003)

59. Axel Gerstenberger and Wolfgang A. Wall, An eXtended Finite Element Method/Lagrange multiplier based approach for fluid–structure interaction, Computer Methods in Applied Mechanics and Engineering, Vol. 197, No. 19-20, P. 1699-1714 (2008)

60. G. Ventura, On the elimination of quadrature sub-cells for discontinuous functions in the extended finite-element method, *International Journal for Numerical Methods in Engineering*, Vol. 66, No. 5, P. 761-791 (2006)

61. H. Ji and J. E. Dolbow, On strategies for enforcing interfacial constraints and evaluating jump conditions with the extended finite element method, *International Journal for Numerical Methods in Engineering*, Vol.61, No. 14, P. 2508-2535 (2004)

62. S.E. Mousavi and N. Sukumar, Generalized Gaussian quadrature rules for discontinuities and crack singularities in the extended finite element method, *Computer Methods in Applied Mechanics and Engineering*, Vol. 199, No. 49-52, P. 3237-3249 (2010)

63. Jeong Ho Kim, Chang Sung Lee and Seung Jo Kim, High-performance domainwise parallel direct solver for large-scale structural analysis, *AIAA Journal*, Vol.43, No 3, P. 662-670 (2005)

64. Soon Wan Chung and Seung Jo Kim, Parallel computation of a damage localization problem using parallel multifrontal solver, Computational Mechanics, Vol. 33, No. 1, P. 42-51 (2003)

65. O.C. Zienkiewicz and R.L. Taylor, The finite element method volume 3: Fluid dynamics, Butterworth Heinemann, 5th edition (2000)

66. Zhong Cai, Analysis of mold filling in RTM process, *Journal of Composite Materials*, Vol. 26, No.9, P. 1310-1338 (1992)

67. Douglas S. Cairns and Scott M. Rossell, Fluid flow modeling of Resin Transfer Molding for composite material wind turbine blade structures, *Sandia National Laboratories* (2004)

68. Douglas S. Cairns, Ladean R. McKittrick, David C. Combs, John F. Mandell, Donald A. Rabern, and R. Daniel VanLuchene, Design/manufacturing synthesis of a composite blade for the AOC 15/50 wind turbine, *18th ASME Wind Energy Symposium*, P. 58-65 (1999)

69. Ladean R. McKittrick, Analysis of a composite blade design for the AOC 15/50 wind turbine using a finite element model, *Sandia National Laboratories* (2001)

70. Seung Jo Kim, Performance Validation of Parallel Finite Element Structural Analysis Code, IPSAP and Its Pre-Post Processor, DIAMOND, *International Conference on Computational & Experimental Engineering and*

Sciences(ICCES'10), LAS VEGAS(USA), 28 March - 1 April, P. 507-507, (2010)

71. Suresh G. Advani and E. Murat Sozer, Process modeling in composites manufacturing, 2nd edition, Cambridge University Press (2003)

72. Mark Lin, H. Thomas Hahn, and Hoon Huh, A finite element simulation of resin transfer molding based on partial nodal saturation and implicit time integration, *Composite Part A: Applied Science and Manufacturing*, Vol. 29, No. 5-6, P. 541-550 (1998)

Appendix A. Multi-frontal solver of IPSAP

1. IPSAP

IPSAP (Internet Parallel Structural Analysis Program) is large scale, high performance structural analysis software developed by Aerospace Structures Laboratory. IPSAP is divided into two types of solution package, IPSAP/Standard and IPSAP/Explicit. IPSAP/Standard is able to solve linear static analysis, thermal conduction analysis, and vibration analysis of structures, and IPSAP/Explicit has ability to solve time-dependent structural problem such as crash, metal forming, and penetration phenomena.

IPSAP/Standard has 3 type solution modules, 2 linear solution methods and 1 eigensolution method. Linear solver in IPSAP/Standard contains both serial/parallel version of Multifrontal method and hybrid domain decomposition iterative method, and eigensolver is block-Lanczos method. Multifrontal solver is the best direct solution method in terms of requirements of computations, memory, and parallel efficiency. It is extremely efficient in the both serial and parallel machines. Hybrid domain decomposition method is based on FETI-DP method which uses multifrontal method to computational efficiency. Last, block-Lanczos eigensolver was developed for multiple eigenvalues/eigenvectors of large scale problems. IPSAP/Standard is published in Windows(x86, x64, serial and parallel), Linux(x86, x64, serial and parallel), Mac OS(G4, G5, serial) version.

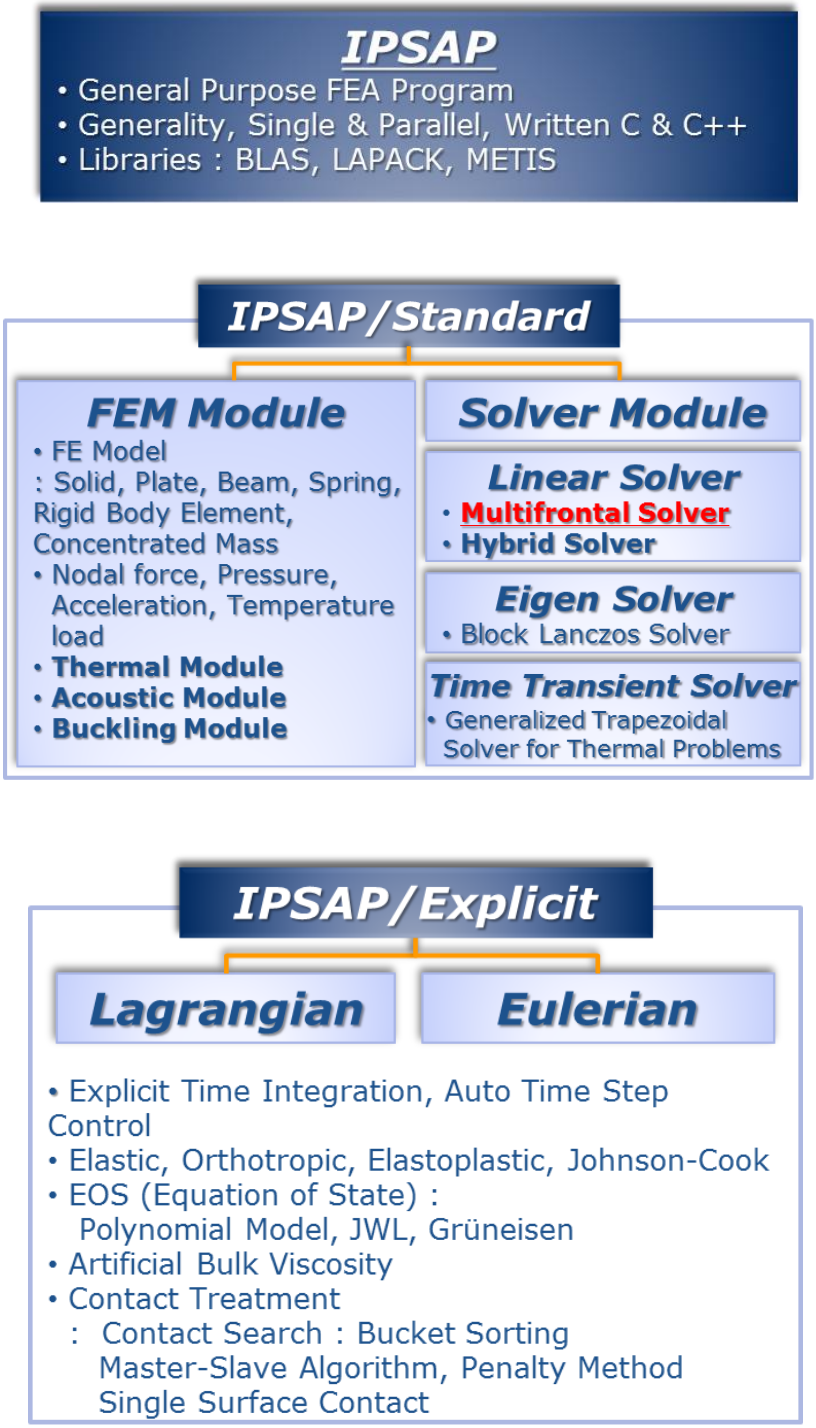
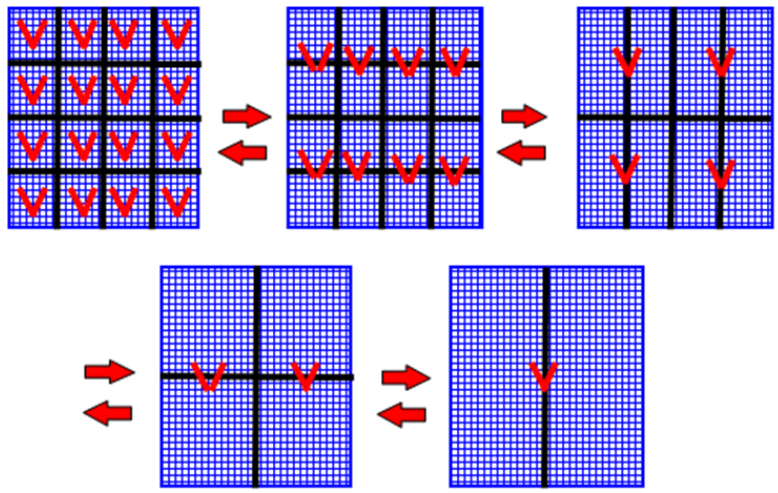


Figure A.1. Main feature of IPSAP

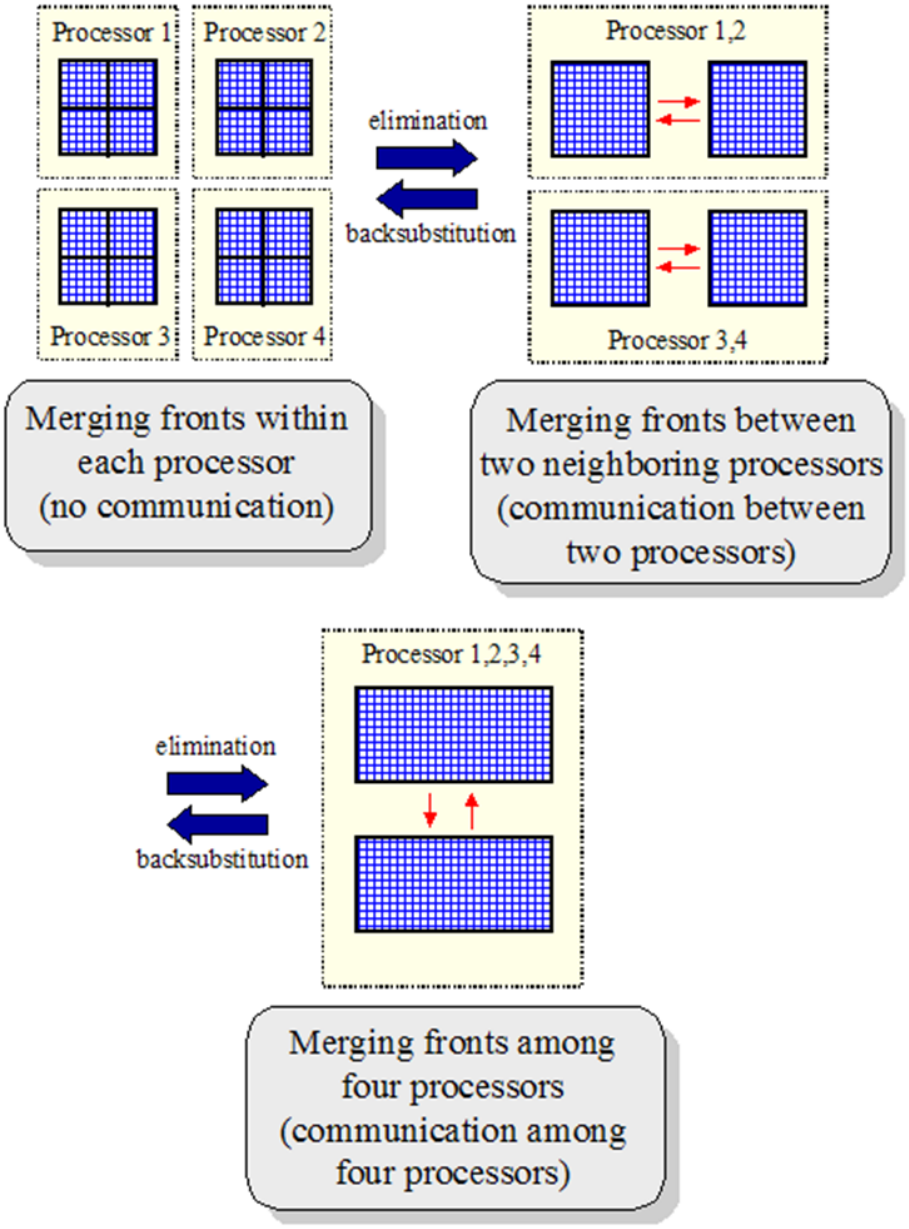
2. Multi-Frontal Solver

Multifrontal method was proposed to minimize requirements of memory and computation for finite element methods. Multifrontal solver in IPSAP is optimized for finite element problems, direct version of domain decomposition methods. Thus, the solver doesn't need to assemble global stiffness matrix, whereas other direct solvers require assembling global stiffness matrix. IPSAP lies on distributed memory parallelism. To implement in parallel computing, PLASC was developed to handle flexible block size of matrix on distributed memory. The general introduction of parallel computing is included in Appendix C.

The figure A.2 illustrates the factorization of MFS and parallel extend-add operation on distributed memory, respectively.



(a) Factorization of Multi-frontal solver



(b) Parallel extend-add operation

Figure A.2. Multi-frontal solver of IPSAP

Appendix B. Key-points of cross-sections for the composite wind turbine blade model

The composite wind turbine blade model is drawn by 10 cross-sections. The key-points of the 10 cross-sections are given in the table 4. The cross-section number is started from the root end.

Table 4. Key-points of 10 cross-sections for the composite wind turbine blade model

Cross Section	x(m)	y(m)	z(m)
1	0.0000	0.3000	0.0000
1	0.1677	0.4264	0.0000
1	0.2258	0.4353	0.0000
1	0.4498	0.4176	0.0000
1	0.5501	0.3564	0.0000
1	0.3411	0.2251	0.0000
1	0.2258	0.1917	0.0000
1	0.1954	0.1871	0.0000
1	0.1354	0.1803	0.0000
2	0.0000	0.3000	0.3983
2	0.1317	0.3991	0.3983
2	0.1982	0.4130	0.3983
2	0.2264	0.4177	0.3983
2	0.3178	0.4180	0.3983
2	0.4989	0.4057	0.3983
2	0.6477	0.3514	0.3983

2	0.5429	0.3143	0.3983
2	0.3142	0.2244	0.3983
2	0.2264	0.1971	0.3983
2	0.1954	0.1905	0.3983
2	0.1305	0.1849	0.3983
3	0.0000	0.3000	1.0633
3	0.1281	0.3882	1.0633
3	0.1960	0.4050	1.0633
3	0.2265	0.4091	1.0633
3	0.3279	0.4112	1.0633
3	0.5989	0.3858	1.0633
3	0.3266	0.2411	1.0633
3	0.2254	0.2035	1.0633
3	0.1953	0.1978	1.0633
3	0.1271	0.1942	1.0633
3	0.7214	0.3516	1.0633
3	0.5907	0.3267	1.0633
4	0.0000	0.3000	1.7363
4	0.1273	0.3803	1.7363
4	0.2272	0.3997	1.7363
4	0.3212	0.4033	1.7363
4	0.5552	0.3858	1.7363
4	0.7493	0.3517	1.7363
4	0.6221	0.3391	1.7363
4	0.3177	0.2374	1.7363
4	0.2272	0.2065	1.7363
4	0.1962	0.1992	1.7363
4	0.1263	0.2031	1.7363
5	0.0000	0.3000	2.4103

5	1.2230	0.3690	2.4103
5	0.1892	0.3837	2.4103
5	0.3215	0.3902	2.4103
5	0.5281	0.3701	2.4103
5	0.7173	0.3417	2.4103
5	0.5254	0.3206	2.4103
5	0.3228	0.2531	2.4103
5	0.2204	0.2209	2.4103
5	0.1897	0.2158	2.4103
5	0.1230	0.2217	2.4103
6	0.0000	0.3000	3.0439
6	0.1162	0.3605	3.0439
6	0.1855	0.3745	3.0439
6	0.2151	0.3779	3.0439
6	0.3047	0.3801	3.0439
6	0.4685	0.3593	3.0439
6	0.6579	0.3304	3.0439
6	0.5530	0.3233	3.0439
6	0.3035	0.2504	3.0439
6	0.2144	0.2309	3.0439
6	0.1851	0.2273	3.0439
6	0.1145	0.2387	3.0439
7	0.0000	0.3000	3.7351
7	0.1097	0.3537	3.7351
7	0.1800	0.3645	3.7351
7	0.2098	0.3680	3.7351
7	0.3141	0.3664	3.7351
7	0.4576	0.3418	3.7351
7	0.6071	0.3213	3.7351

7	0.4946	0.3099	3.7351
7	0.3128	0.2578	3.7351
7	0.2078	0.2404	3.7351
7	0.1773	0.2421	3.7351
7	0.1078	0.2542	3.7351
8	0.0000	0.3000	4.4263
8	0.1035	0.3458	4.4263
8	0.1707	0.3564	4.4263
8	0.2017	0.3598	4.4263
8	0.3074	0.3579	4.4263
8	0.4464	0.3328	4.4263
8	0.5537	0.3134	4.4263
8	0.4725	0.3056	4.4263
8	0.3060	0.2660	4.4263
8	0.2013	0.2514	4.4263
8	0.1697	0.2542	4.4263
8	0.1037	0.2640	4.4263
9	0.0000	0.3000	5.0923
9	0.0918	0.3399	5.0923
9	0.1659	0.3477	5.0923
9	0.1959	0.3503	5.0923
9	0.2991	0.3471	5.0923
9	0.3931	0.3263	5.0923
9	0.5004	0.3059	5.0923
9	0.3904	0.2927	5.0923
9	0.2960	0.2737	5.0923
9	0.1954	0.2619	5.0923
9	0.1648	0.2637	5.0923
9	0.0910	0.2715	5.0923

10	0.0000	0.3000	5.7573
10	0.0899	0.3345	5.7573
10	0.1583	0.3412	5.7573
10	0.1899	0.3414	5.7573
10	0.2921	0.3354	5.7573
10	0.3643	0.3198	5.7573
10	0.4445	0.3046	5.7573
10	0.3741	0.2983	5.7573
10	0.2915	0.2847	5.7573
10	0.1893	0.2758	5.7573
10	0.1584	0.2746	5.7573
10	0.0892	0.2768	5.7573

Appendix C. Introduction to Parallel Computing

1. Serial vs. Parallel

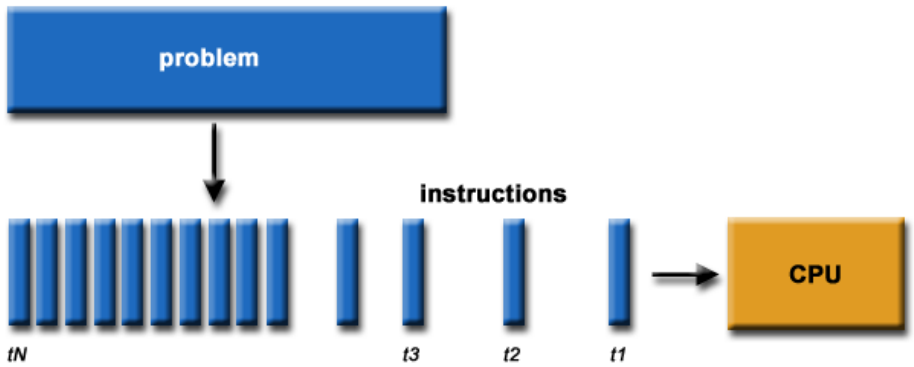


Figure C.1. Description of serial computing

Serial computing is to use a single CPU (processor) to complete a single task, as shown in the figure C.1. The methods to improve the performance of serial computing are to optimize the program or improve the hardware (Moore's law).

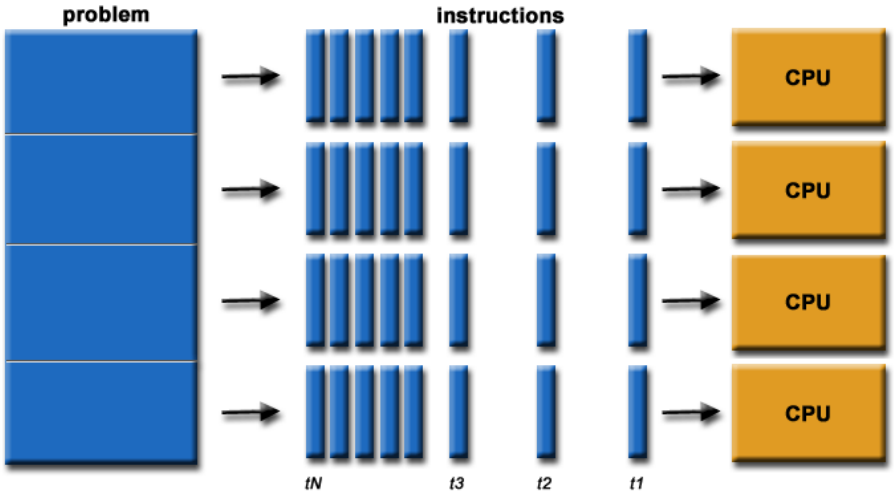


Figure C.2. Description of parallel computing

On the other hand, as shown in the figure C.2, parallel computing is to use the multiple CPUs to reduce the time needed to complete a single task.

2. Classification of computer architectures

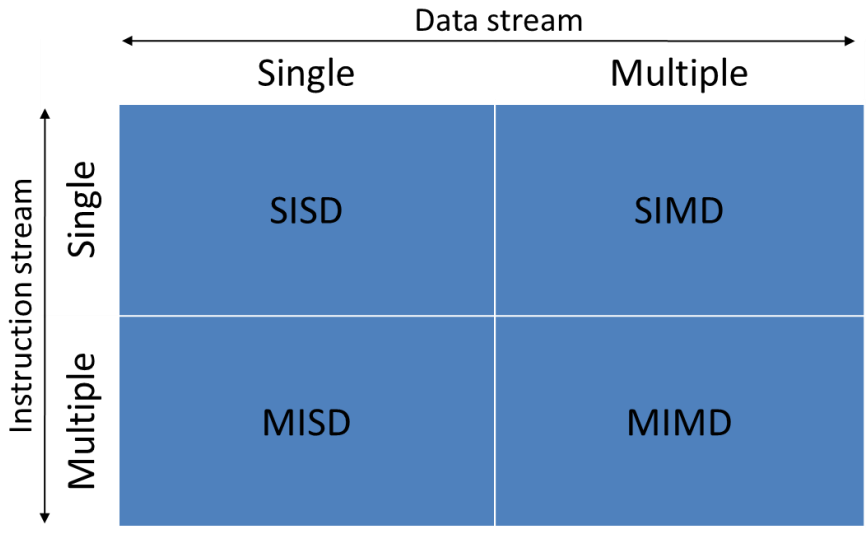


Figure C.3. Classification by Flynn's taxonomy

The Flynn's taxonomy classifies the computer architectures to four divisions: SISD, SIMD, MISD, and MIMD. The classification is described in the figure C.3.

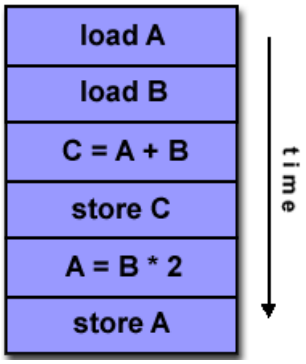


Figure C.4. Operation in SISD

SISD is an acronym of a Single Instruction stream and a Single Data stream. A serial computer is included in this division. As shown in the figure C.4, only one instruction stream and on data stream are executed.

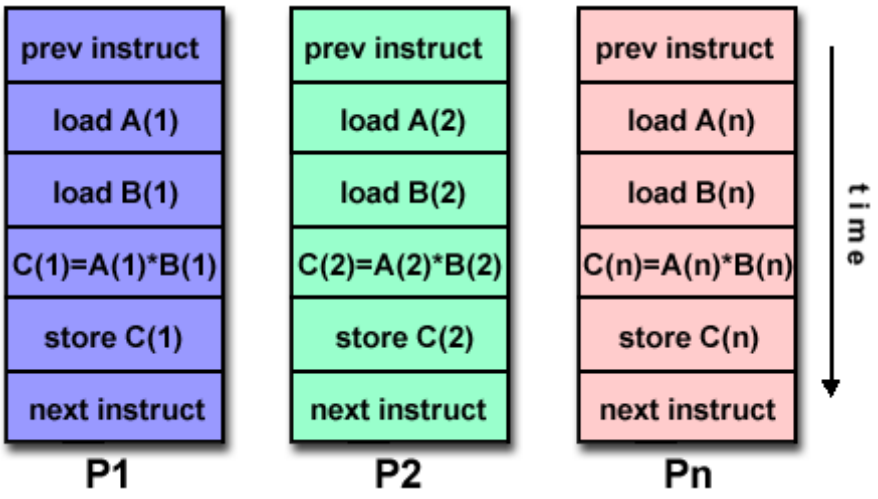


Figure C.5. Operation in SIMD

SIMD is an acronym of a Single Instruction stream but Multiple Data streams. All processing units execute the same instruction. And, each processing unit can operate on a different data element as the figure C.5. The early supercomputers such as Thinking Machines CM-1 and CM-2 are included in this division.

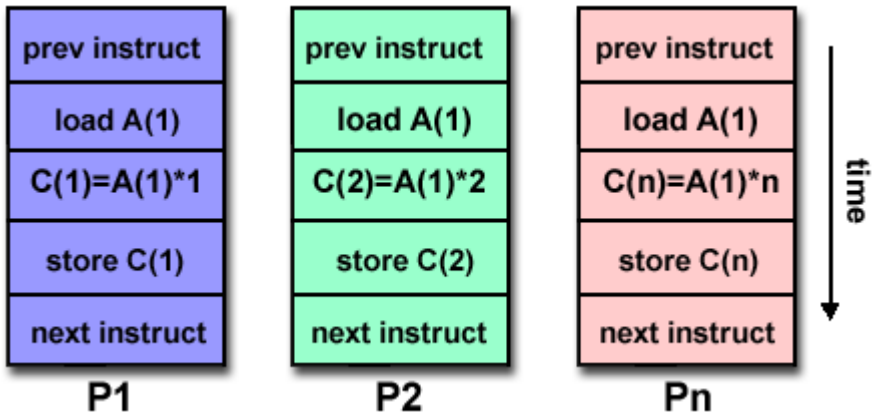


Figure C.6. Operation in MISD

MISD is an acronym of Multiple Instruction streams and a Single Data stream as the figure C.6. In MISD computers, various instructions operate with the same data. Few actual examples of this class of parallel computer have existed.

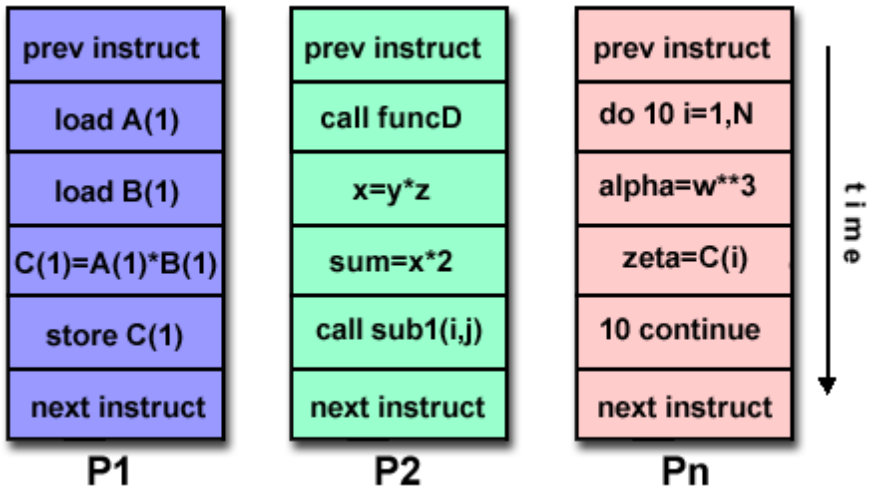
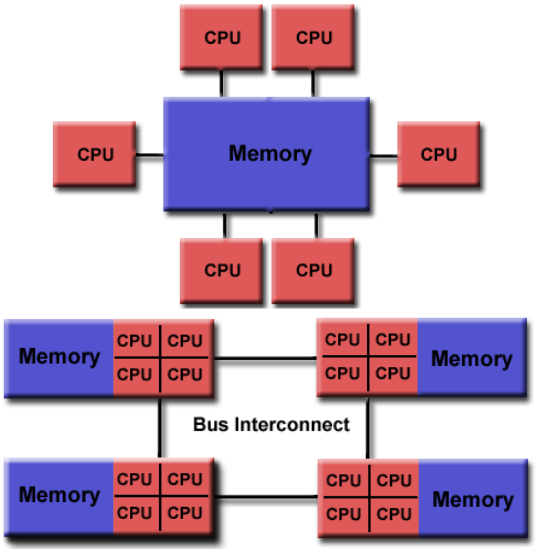


Figure C.7. Operation of MIMD

MIMD is an acronym of Multiple Instruction streams and Multiple Data streams. The operation of MIMD is described in the figure C.7. The most common type of parallel computers is included in this division. Every processor executes a different instruction stream with a different data stream.

3. Parallel computer memory architectures



(a) UMA (b) NUMA

Figure C.8. Architecture of shared memory

The memory architectures of parallel computer are classified to shared, distributed, and hybrid distributed-shared. In shared memory system, there are Uniform Memory Access (UMA) and Non-uniform Memory Access (NUMA). The architectures of UMA and NUMA is described in the figure C.8. Advantage of shared memory is that the data access of CPUs is fast due to the closeness between the memory

and the CPUs. And, disadvantage of it is the expensive cost to increase the size of the memory and the number of CPUs.

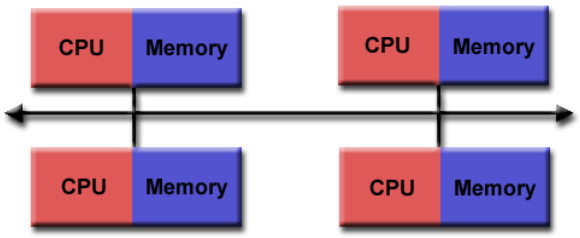


Figure C.9. Distributed memory

The figure C.9 shows the architecture of distributed memory. In this case, each processor has its own local memory. The advantage is the inexpensive cost to increase the number of processors and the size of memory. In addition, each processor can rapidly access to its own memory. The disadvantage is that the data communication between CPUs is slower than the shared memory.

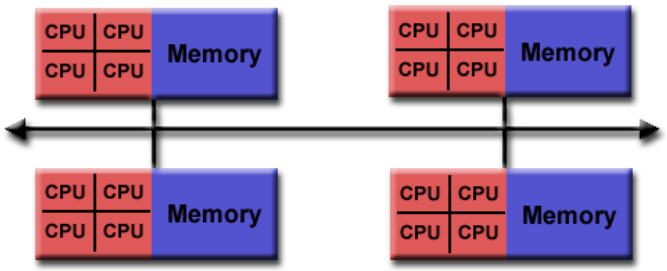


Figure C.10. Hybrid distributed-shared memory

The last architecture is hybrid distributed-shared memory as shown in the figure C.10. The largest and fastest computers in the world today employ both shared and distributed memory architectures (cluster machines with multi-core CPUs).

4. Parallel programming

The speedup of a program using multiple processors in parallel computing is limited by the sequential fraction of the program. The speedup is calculated by Amdahl's law:

$$\text{Speedup} = \frac{1}{1 - P + \frac{P}{N}} \tag{35}$$

, where, P is parallel portion, N number of processors.

Application Programming Interfaces (API) of Parallelization are MPI and OpenMP. MPI is an acronym of Message Passing Interface. It is functions to specify message-passing for the parallel programming in Fortran or C. OpenMP is API for parallel programming on the shared memory environment.

초 록

확장 유한 요소 법을 적용한 Resin Transfer Molding 기법의 효율적 수치 해석

정연희
협동과정 계산과학 전공
서울대학교 대학원

Resin Transfer Molding (RTM) 공정을 수치해석하기 위해 Level set 방법과 결합된 확장 유한 요소 법을 적용하였다. 유동 전면 부에서 비연속적인 구배를 가지는 압력을 계산하기 위해 확장 유한 요소 법을 이용하여 계산의 정밀성을 높였다. 확장 유한 요소 법에 이용되는 확장 형상 함수는 Level set 값을 이용하여 정의하였다. 이 확장 형상 함수는 요소를 통과하는 수지 유동 전면부의 위치를 반영할 수 있다. 게다가 Level set 법이 금형 충전 동안 수지 유동 전면부의 위치를 계산할 때 적용되었다. 수지 유동 전면부의 위치를 계산하는 미분방정식은 내연적 특성 Galerkin 유한요소법을 적용하여 풀었다. 선형 시스템 계산에서는 IPSAP의 다중 프론트 솔버를 이용한다. 본 연구에서 계산한 해석 값은 이론 값과 비교하여 검증하였다.

계산 효율을 높이기 위해 확장 유한 요소 법과 Level set 방법의 국소화 기법이 제안되었다. 이 기법은 계산 영역을 수지 유동 전면 부 근처의 영역으로 축소한다. 그러므로 전체 계산 양은 최소화될 수 있었다. 이 기법의 계산 효율은 채널과 방사상 유동 모델을 이용하여 평가된다.

본 연구의 해석 능력을 보여주기 위해 몇 가지 적용 예제를 계산하였다. 첫 번째 예제를 이용해서 복잡하게 흘러가는 수지 전면부의 갈라짐과 합쳐지는 현상 해석하였다. 그리고 금형 내부의 Race-tracking 효과와 기공 생성 현상을 확인하기 위해 복잡한 모양의 구조물을 시뮬레이션 하였다. 이와 더불어 풍력 터빈 블레이드 예제를 이용하여 금형 충전 과정을 해석하였다. 마지막으로 제안한 알고리즘은 시뮬레이션의 전/후처리를 위한 그래픽 사용자 인터페이스 툴로서 실현되었다.

주요어 : RTM 공정, 확장유한요소법, Level set 방법, 수치해석, 국소화 기법
학 번 : 2005-23272